Common Criteria for Information Technology Security Evaluation

Smart Card Security User Group
Smart Card Protection Profile

Draft

Draft Version 2.0

May 1, 2000

This protection profile was developed to identify and set forth a comprehensive list of smart card security requirements based on the ISO Standard 15408, the “Common Criteria” (available at http://www.csrc.nist.gov/cc). It is a product of the efforts of the Smart Card Security User Group (SCSUG), a working group formed specifically to represent the security needs of the user community. The members of the SCSUG at the time of this revision included:

- American Express
- Europay International
- JCB Co Ltd
- MasterCard International
- Mondex International
- Visa International
- National Institute of Standards and Technology (United States of America)
- National Security Agency (United States of America)

Ray-McGovern Technical Consultants, Inc. assisted in the preparation of this protection profile.

Addresses and Points of Contact are listed in Annex E.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Identification

Version Number: Draft Version 2.0

Registration: <to be filled in upon registration>

A glossary of terms used in the protection profile (PP) is given in Annex A. This protection profile is hereafter referred to as the Smart Card Security User Group Smart Card Protection Profile (SCSUG-SCPP).

This PP has been built with Common Criteria (CC) Version 2.1 (ISO/IEC 15408-1: 1999 (E), Information technology - Security techniques -- Evaluation criteria for IT security) and Common Methodology for Information Technology Security Evaluation (CEM-97/017, Part 1 : Introduction and General Model, Version 0.6, 97/01/11 and CEM-99/045, Part 2: Evaluation Methodology, Version 1.0, August 1999)

The structure for this PP was established through reference to ISO/IEC PDTR 15446, Information Technology – Security Techniques – Guide for the Production of Protection Profiles and Security Targets, Version 0.9, 2000-01-04, and the Common Criteria Toolbox. This toolbox was developed by SPARTA, Inc., for the US National Security Agency. It is available through http://cctoolbox.sparta.com.

A product compliant with this PP may offer security features and functionality beyond those specified in this PP.

1.2 PP Overview

This PP describes the IT security requirements for a smart card to be used in connection with sensitive applications, such as banking industry financial payment systems. Smart card as used in this PP means an integrated circuit containing a microprocessor, volatile and non-volatile memory, and associated software, packaged and embedded in a carrier. The integrated circuit is a single chip incorporating CPU, RAM, ROM, and programmable non-volatile memory (usually EEPROM). The carrier is typically made of plastic and usually conforms to ISO 7810 and 7813 - Identification Cards, but may have the smaller size of a GSM (global system for mobile communications) subscriber identification module (SIM). The chip is embedded in a module incorporating the commu nication channels (with contacts in accordance with ISO 7816 or contactless in accordance with ISO 14443).

The requirements cover the smart card's integrated circuit and operating software, but do not include specific applications. This PP is applicable to both contact and contactless smart cards, without special regard for form factor or physical card security features. This PP does not cover security requirements for card terminals or networks interfacing with them. It is anticipated that application-specific PPs or security targets could be developed that would incorporate the requirements in this PP as their foundation.

In addition to the security requirements specified in this protection profile, individual applications may have additional security requirements specified in their own protection profiles.

1.3 Assurance Level

The assurance level for this protection profile is EAL4 augmented. Augmentation results from the selection of:

AVA_VLA.3 Vulnerability Assessment - Vulnerability Analysis - Moderately resistant
and
ADV_INT.1 Development - TSF internals - Modularity.

Strength of function is high.

1.4 Related Standards and Documents

ISO 7810 - Identification Cards - Physical Characteristics

ISO 7813 - Identification Cards - Financial Transaction Cards

ISO 7816 - Identification Cards - Integrated Circuit Cards with Contacts

ISO 10202 - Financial Transaction Cards - Security Architecture of Financial Transaction Systems using Integrated Circuit Cards

ISO 14443 (Draft) - Contactless Integrated Circuit Cards, Proximity Cards

ISO 15408 - Information Technology - Security Techniques - Evaluation Criteria for IT Security (Hereafter referred to as Common Criteria or CC)

Common Methodology for Information Security Evaluation (CEM)

1.5 Related Protection Profiles and Documents

This protection profile has evolved from a great deal of work on smart card security conducted over the past decade. Much of this work has been done in conjunction with a variety of organizations (including the Smart Card Forum, Smart Card Industry Association, and Eurosmart), semiconductor and smart card manufacturers, and more than a dozen commercial evaluation laboratories. In particular it has evolved from:

- Visa Smart Card Protection Profile, Draft Version 1.6
- Protection Profile 9806 - Smartcard Integrated Circuit (revision of PP 9704 - Smartcard Integrated Circuit)

- Protection Profile 9810 - Smartcard Embedded Software
- Protection Profile 9911 - Smart Card Integrated Circuit with Embedded Software (supersedes PP9809 - Smart Card Integrated Circuit with Embedded Software)

The Visa Smart Card Protection Profile, Draft Version 1.6 is available at http://www.visa.com and http://csrc.nist.gov/cc/pp/pplist.htm. The protection profiles PP 9806, PP9810, and PP9911 are available at http://www.eurosmart.com and http://www.scssi.gouv.fr. Additional input was obtained from security documents furnished by MasterCard International, Europay International, Visa International, and Mondex International and from multiple open reviews resulting in many comments by a wide range of sources.

Future versions of this document will reduce differences, leading to a unified understanding and approach to smart card security.

While the Smart Card Security User Group gratefully acknowledges the assistance provided by others, the responsibility for this protection profile rests with the SCSUG.

1.6 PP Organization

The main sections of the PP are the TOE (target of evaluation) description, TOE security environment, security objectives, IT security requirements, rationale, and annexes.

The TOE description provides general information about the TOE, serves as an aid to understanding its security requirements, and provides context for the PP's evaluation.

The TOE security environment describes security aspects of the environment in which the TOE is to be used and the manner in which it is to be employed. The TOE security environment includes:
a) assumptions regarding the TOE's intended usage and environment of use
b) threats relevant to secure TOE operation
c) organizational security policies with which the TOE must comply

The security objectives reflect the stated intent of the PP. They pertain to how the TOE will counter identified threats and how it will cover identified organizational security policies and assumptions. Each security objective is categorized as being for the TOE or for the environment.

The security requirements section provides detailed requirements, in separate subsections, for the TOE and its environment.

The IT security requirements are subdivided as follows:
a) TOE Security Functional Requirements
b) TOE Security Assurance Requirements

The rationale presents evidence that the PP is a complete and cohesive set of requirements and that a conformant TOE would provide an effective set of IT security countermeasures within the security environment. The rationale is in two main parts. First, a security objectives rationale demonstrates that the stated security objectives are traceable to all of the aspects identified in the TOE security environment and are suitable to cover them. Then, a security requirements rationale demonstrates that the security requirements (TOE and environment) are traceable to the security objectives and are suitable to meet them.

The annexes constitute application notes for this PP. In addition to a glossary and points of contact, the annexes provide supporting information on issues unique to smart cards, consideration of management functions, and suggestions for the application of this PP through the use of packages applying to the basic chip, operating software, and integrated platform.

2 TOE Description

2.1 Smart Card Overview

A smart card or integrated circuit card (ICC) is a computer chip embedded into a carrier. The chip is a semiconductor (silicon) integrated circuit (IC) fabricated in a complex microelectronic process, which involves repeatedly masking and doping the surface of a silicon substrate to form transistors, followed by patterning metal connections, and applying a protective overcoat. This process eventually yields a design typically comprising several hundred thousand transistors, arranged in an area less than 25 square millimeters. The design consists of a central processing unit, an optional co-processor, input and output lines, and volatile and non-volatile memory.

The chip will also be designed to be secure. In order to be secure, it should make appropriate use both of specific design features that are dedicated to security, e.g. environmental sensors, and also of technological properties of the materials and processes used.

A part of the manufacturing process is the inclusion of operating software (OS). This is developer-specific code, written in the microprocessor’s native or machine code. Operating software is usually contained in one of the numerous masks used during manufacture, referred to in this document as the ROM mask.

The IC itself is packaged. The current predominant method is die bonding in a module. A module consists of a small board on which the IC is seated. Wire bonds are connected from the IC’s input/output (I/O) pads to the carrier, which has contacts on its reverse side. The chip is then encapsulated in a protective material (usually some type of epoxy) and the module is adhesively embedded into a pre-milled hole in the plastic card. Two common examples are the familiar payment card-sized smart cards and the smaller postage stamp-sized subscriber identity module (SIM) frequently used in mobile telephones.

Additional information regarding the life cycle of smart cards in general is presented in Annex B.

2.2 Definition of TOE

The target of evaluation (TOE) for this protection profile is an operational smart card platform, consisting of the integrated circuit and operating software, including the mechanisms that allow communication with the outside world. The TOE consists of sufficient hardware and software elements to be capable of establishing a secure channel to a trusted source for application loading or for other potentially privileged commands.

This PP does not include printing, the magnetic stripe (if present), security features such as holograms, or any other part of the card. This protection profile also does not apply to the card accepting device (terminal), nor to any network with which the integrated circuit card interfaces.

The TOE is intended to be suitable for use in financial services systems, but is not limited to that application. Detailed information reflecting the security needs of any particular application would impose requirements which would not, in general, be appropriate for other applications. It is therefore anticipated that this protection profile will be supplemented with an application specific protection profile. The author of the application specific protection profile is responsible for detailing the additional security requirements necessary for instantiation of a fully capable smart card system for that specific application.

2.3 TOE Identification

Through selection of the ACM Configuration Management Class of assurance functions, this PP imposes the requirement that a unique reference be utilized to ensure that there is no ambiguity in terms of which instance of the TOE is being evaluated. Labeling the TOE with this reference then ensures that users of the TOE can be aware of which instance of the TOE they are using. The TOE described herein is, however, a combination of hardware and software, each portion of which may be composed of a further collection of components. This aggregate collection offers the potential for confusion in identifying a unique reference for the TOE.

To further complicate identification, commonly an IC can be produced with multiple features, only some of which are enabled. The design layout of the IC (the photomask) determines the functionality; however, as fabrication technology improves, the identical design may be used to produce an otherwise identical chip but with a reduced feature size. Likewise, software features may be selectively employed, depending on hardware functions. Moreover, the presence or absence of specific features may directly contribute to the possible introduction of vulnerabilities. For example, the size of the IC features is directly related to the relative difficulty of probing. A potentially unknown, but present, software feature may allow backdoors or other routes for penetration.

It is therefore essential that the unique reference for the TOE compliant with this PP allow the identification of at least:

- the microprocessor specification
- the memory size and allocation (ROM, EEPROM, RAM, etc.)
- the physical instantiation of the IC design regarding layout and feature size
- all hardware security features on the IC, whether they are initially enabled or not
- all enabled hardware security features
- the software specification
- all software security features present, whether they are initially enabled or not
- all enabled software security features

2.4 Cryptography

A variety of cryptographic keys are typically used with smart cards, including transport keys, personalization keys, application-specific keys, etc. Handling of these keys must be done in accordance with the key management procedures and policies of the issuing organization.

Cryptography may be implemented in hardware or software, with various algorithms and various key lengths. Many smart cards have dedicated crypto coprocessors that execute DES, triple DES, RSA and other standard algorithms much faster than software implementations can. Some applications use no cryptography, some use private key, and some public key systems.

Any TOE claiming compliance with this protection profile must handle cryptographic functions in accordance with applicable international, industrial, or organizational policies. This extends to any applications using cryptography, although there may be additional applications on the card that do not use cryptography at all.

2.5 Environments

Smart card environments are highly variable and to some extent application dependent. In general, a smart card is assumed to be in the uncontrolled possession of the cardholder. The card must therefore protect its assets against unauthorized alteration that may be accomplished with standard personal computers and with laboratory equipment used without any supervision. Typically, the cards are designed for world-wide use in a wide variety of card acceptance devices ranging from parking meters and vending machines to dedicated read/write devices or card readers attached to conventional computers.

2.6 Attacker Capabilities

Attackers are assumed to have various levels of expertise, resources, and motivation. Relevant expertise may be in general semiconductor technology, software engineering, hacking techniques, or in the specific TOE. Resources may range from personal computers and inexpensive card reading devices to very expensive and sophisticated engineering test and measurement devices. They may also include software routines, some of which are readily available on the Internet. Motivation may include economic reward or the satisfaction and notoriety of defeating expert security. It is assumed that given sufficient time and expertise, any smart card can be compromised.

It is imperative that security targets and smart card products claiming compliance with the SCSUG-SCPP be clearly identified as to type of mask programming being utilized and that security functions that are present are appropriate to that type of card.

2.7 Reader

The card reader (card acceptor device or CAD) provides the interface between the smart card and the rest of the environment. It provides power and clock to the smart card. The reader generates a reset signal and applies it to the correct port on the card. The reader then links to the input/output port to provide all communications to and from the card.

The simplest CAD may be a value checker for a stored value application. In this function, the card holder would insert the smart card into the reader, which would perform a simple query, returning the amount of value remaining on the card on a small alphanumeric display. Such devices might also be able to read activity logs, but would not have write capability. No other functions would be possible with this type of CAD.

More complex CADs have write capability and could include additional output devices such as larger displays, printers, or connections to networks. Input devices, including PIN pads or keyboards, could be present. The CAD could contain memory and computational components as well. Typically the CAD has an internal program that interfaces with a program on the smart card and both programs are required to conduct a transaction.

Cryptographic functions may be necessary in the CAD to support certain applications. These may include storing secret or private keys; providing cryptographic operations such as encryption, digital signature or hashing; or processing secure card data for transmission over a network connected to the CAD. In order to maintain the security of these operations, the CAD typically would be equipped with a security module providing protection to this information. The requirements for such a security module are appropriate topics for a protection profile but are outside the work presented here.

The location of CADs can not be reasonably assumed. In the case of value checkers, the intent is that the reader be readily accessible to the card holder, perhaps being carried in a pocket. Readers associated with mobile phones would, of necessity, be correspondingly mobile. CADs providing access through personal computers could be home or office based. In many systems, the CAD may be owned and operated by someone other than the card holder, e.g., a merchant, doctor, pharmacist, etc.

It is thus seen that CADs must be considered an uncontrolled item, so the security provided by the CAD can not be assumed (other than as addressed in a well defined security module as mentioned above).

3 TOE Security Environment

This section identifies the following:

- Identification of assets
- Significant assumptions about the TOE's operational environment
- IT-related threats to the organization countered by SCSUG-SCPP compliant components
- Threats requiring reliance on environmental controls to provide sufficient protection
- Organizational security policies for which SCSUG-SCPP compliant TOEs are appropriate

3.1 Assets

The primary asset of concern to this PP is the user data representing information to be protected. In the context of this TOE, user may be defined as the ultimate end user (e.g., the card holder) or as an application which is loaded onto the completed TOE. In the first case, the user is associated with specific data which supports the functions the TOE is performing for that user. In the second case, the user is the new application resident on the card. All code and operations of that application are considered user functions. The final user is not visible to the TOE operations. Thus, the primary asset of user data may be considered to be either (or both of):

- card holder data in support of direct TOE functions
- application code which is added to the completed TOE to incorporate further functionality

Certain data is required to support the secure operations regarding the above defined user data. This TSF data includes:

- security attributes, authentication data and access control list entries
- the various cryptographic keys which are used in the security processes of the TOE

The use characteristics of smart cards require them to be in the hands of users for prolonged periods of time. As discussed elsewhere in this PP, this can be considered a hostile environment. It is therefore necessary to consider the protection of those characteristics of the TOE and its design that support the preservation of security for the primary assets. The secondary assets of concern to this PP therefore include:

- the IC design and specifications
- the software design and specifications, implementation, and related documentation
- the IC and software development tools and technology

Assets are to be protected in terms of confidentiality and integrity.

3.2 Assumptions

The specific conditions listed below are assumed to exist in the smart card environment. Each assumption is stated in bold type font. It is followed by an application note, in normal font, which supplies additional information and interpretation.

A.CAD_Sec-Com - Card Acceptor Device Secure Communication

A CAD to which the TOE establishes a secure link is assumed to be secure.

The CAD may have the capability to establish a secure communication channel with the TOE. This is typically accomplished through shared private keys, public/private key pairs, and/or generation of session keys derived from other stored keys. It is assumed that when such a secure link is established, the TOE may consider the CAD to be adequately secure for trusted communications. The CAD is considered to be beyond the scope of this PP.

A.Data_Store - Off-TOE Data Storage

Management of TOE data off of the TOE is assumed to be performed in a secure manner.

Significant information regarding TOE profile, personalization, ownership, etc. may be held by issuers or others in data bases not associated with the TOE. This information could contribute to a cloning attack. It is therefore important that the security of such data be adequately maintained.

A.Key_Supp - Key Support

All imported cryptographic keys are assumed to be supported off-card in a secure manner.

A variety of keys may be imported for use by, and in conjunction with, the TOE. These may include shared private keys, public/private key pairs, etc. These keys will be supplied from the various bodies controlling the operations of the system in which the TOE is functioning. It is assumed that the generation, distribution, maintenance, and destruction of these keys is adequately secure.

A.Pwr_Clock - Power and Clock

Power and clock come from the CAD. These are not considered reliable sources.

The TOE is internally unpowered, so support must be delivered to the card from the card acceptor device or through an alternate connection to the TOE terminals. Both power and clock may be interrupted or reset in the normal course of business. The CAD is independent of the TOE and may belong to a different entity which may be considered in some way hostile. Power may deviate from the design level (above or below) and may be supplied intermittently. The clock can likewise be manipulated.

A.Role_Man - Role Management

Management of roles for the TOE is performed in a secure manner off-card.

The various roles involved in working with the TOE are established in the development and user community through the TOE manufacturers, card issuing bodies, etc. These roles will be managed off-card by these or other appropriate bodies.

3.3 Threats

SCSUG-SCPP compliant TOEs are required to counter threats that may be broadly categorized as:

- Threats addressed by the TOE
  - Threats associated with physical attack on the TOE
  - Threats associated with logical attack on the TOE
  - Threats associated with control of access
  - Threats associated with unanticipated interactions
  - Threats regarding cryptographic functions
  - Threats which monitor information
  - Miscellaneous threats
- Threats addressed by the Operating Environment

Each threat is stated in bold type font. It is followed by an application note, in normal font, which supplies additional information and interpretation.

3.3.1 Threats Addressed by the TOE

3.3.1.1 Threats Associated with Physical Attack on the TOE

T.P_Probe - Physical Probing of the IC

An attacker may perform physical probing of the TOE to reveal design information and operational contents.

Such probing may include electrical functions but is referred to here as physical since it requires direct contact with the chip internals. Physical probing may entail reading data from the chip through techniques commonly employed in IC failure analysis and IC reverse engineering efforts. The goal of the attacker is to identify such design details as hardware security mechanisms, access control mechanisms, authentication systems, data protection systems, memory partitioning, or cryptographic programs. Determination of software design, including initialization data, personalization data, passwords, or cryptographic keys may also be a goal.

T.P_Modify - Physical Modification of the IC

An attacker may physically modify the TOE in order to reveal design or security related information.

This modification may be achieved through techniques commonly employed in IC failure analysis and IC reverse engineering efforts. The goal is to identify such design details as hardware security mechanisms, access control mechanisms, authentication systems, data protection systems, memory partitioning, or cryptographic programs. Determination of software design, including initialization data, personalization data, passwords, or cryptographic keys may also be a goal.

T.E_Manip - Electrical Manipulation of the IC

An attacker may utilize electrical probing and manipulating of the TOE to modify security critical data so that the TOE can be used fraudulently.

This modification may include manipulation of debug lockouts, first use indicators, card use blocking, blocking function configuration, card block indicators, or card disablement indicators. This threat is distinguished by the intent to utilize a modified TOE rather than to derive information from the TOE.

3.3.1.2 Threats Associated with Logical Attack on the TOE

T.Flt_Ins - Insertion of Faults

An attacker may determine security critical information through observation of the results of repetitive insertion of selected data.

Insertion of selected inputs followed by monitoring the output for changes is a relatively well known attack method for cryptologic devices that can be applied to this TOE as well. The intent is to determine user and TSF related information based on how the TOE responds to the selected inputs. This threat is distinguished by the deliberate and repetitive choice and manipulation of input data as opposed to random selection or manipulation of the physical characteristics involved in input/output operations.

T.Forcd_Rst - Forced Reset

An attacker may force the TOE into a non-secure state through inappropriate termination of selected operations.

Attempts to generate a non-secure state in the TOE may be made through premature termination of transactions or communications between the TOE and the card reading device, by insertion of interrupts, or by selecting related applications that may leave files open.

T.Inv_Inp - Invalid Input

An attacker or authorized user of the TOE may compromise the security features of the TOE through introduction of invalid inputs.

Invalid input may take the form of operations which are not formatted correctly, requests for information beyond register limits, or attempts to find and execute undocumented commands. The result of such an attack may be a compromise in the security functions, generation of exploitable errors in operation, or release of protected data.

T.Load_Mal - Data Loading Malfunction

An attacker may maliciously generate errors in set-up data to compromise the security functions of the TOE.

During the stages of card preparation which involve loading the TOE with special keys, identification of roles, etc., the data itself may be changed from the intended information or may be corrupted. Either event could be an attempt to penetrate the TOE security functions or to expose the security in an unauthorized manner.

T.Reuse - Replay Attack

An unauthorized user may penetrate the TOE through reuse of previously valid authentication data.

Attempts to replay a completed (or partially completed) operation may be used in an attempt to bypass security mechanisms or to expose security-related information.

T.Search - Data Space Search

An attacker may utilize a repeated search of the data space to identify critical information.

Repetitive read commands can be used to attempt the extraction of secure information. This threat is distinguished by the use of valid commands with valid range requests that are repeated to reveal as much as possible of the data space.

T.UA_Load - Unauthorized Program Loading

An attacker may utilize unauthorized programs to penetrate or modify the security functions of the TOE.

Unauthorized programs may include the execution of legitimate programs not intended for use during normal operation or the unauthorized loading of programs specifically targeted at penetration or modification of the security functions.

3.3.1.3 Threats Associated with Control of Access

T.Access - Invalid Access

A user or an attacker of the TOE may access information or resources without having permission from the person who owns or is responsible for the information or resources.

Each authorized role has certain specified privileges which allow access only to selected portions of the TOE and its contained information. Access beyond those specified privileges could result in exposure of secure information.

T.First_Use - Fraud on First Use

An attacker may gain access to TOE information by unauthorized use of a new, previ ously unissued TOE.

The process of issuance may involve setting of indicators in the TOE or notification by the TOE to the (external) issuing bodies that this specific TOE is now in operation. Attempts to use an unissued TOE without such mandated approval could result in fraudulent use.

T.Impers - Impersonation

An attacker may gain access to TOE information by impersonating an authorized user of the TOE.

The TOE is required to allow certain roles to be granted certain privileges. Impersonation of a user with such privileges could expose security functions or information which is to be protected by the TOE from unauthorized release.

3.3.1.4 Threats Associated with Unanticipated Interactions

T.App_Ftn - Use of Unallowed Application Functions

An attacker may exploit interactions between applications to expose sensitive TOE or user data.

Interactions may include execution of commands that are not required or allowed in the specific application being performed. Examples include use of native COS functions that are unnecessary or that could compromise security. Inappropriate interactions could also include passing secure information such as PINs or cryptographic data between applications, or transferring value or information into applications that have been exited.

T.LC_Ftn - Use of Unallowed Life Cycle Functions

An attacker may exploit interactions between life-cycle functions to expose sensitive TOE or user data.

Interactions may include execution of commands that are not required or allowed in the specific phase of operation being executed. Examples include use of test, debug or native COS functions that are unnecessary or that could compromise security.

T.Res_Con - Resource Contention

A user or attacker may willfully, or though negligence, monopolize resources of the TOE denying service to another user.

If the limited resources of the TOE are allocated to a user or attacker without the authorization of the owner of the resource, then another that which requires the same resource may not be able to operate normally.

3.3.1.5 Threats Regarding Cryptographic Functions

T.Crypt_Atk - Cryptographic Attack

An attacker may defeat security functions through a cryptographic attack against the algorithm or through a brute-force attack.

This attack may include either encode/decode functions or random number generators.

3.3.1.6 Threats which Monitor Information

T.I_Leak - Information Leakage

An attacker may exploit information which is leaked from the TOE during normal usage.

Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. This leakage may be interpreted as a covert channel transmission but is more closely related to measurement of operating parameters, which may be derived either from direct (contact) measurements or measurement of emanations and can then be related to the specific operation being performed.

T.Link - Linkage of Multiple Observations

An attacker may observe multiple uses of resources or services and, by linking these observations, deduce information that would reveal critical security information.

The combination of observations over a period of many uses of the TOE or the integration of knowledge gained from observing different operations may reveal information that allows an attacker to either learn information directly or to formulate an attack that could further reveal information that the TOE is required to keep secret.

3.3.1.7 Miscellaneous Threats

T.Env_Strs - Environmental Stress

An attacker may exploit failures in the TOE induced by environmental stress.

Exposure of the integrated circuit to conditions outside its specified operating range may result in malfunction or failure of security critical components, allowing manipulation of programs or data. These conditions could either be extremes (high or low) in normal parameters such as temperature, voltage, or clock frequency, or could be abnormal conditions such as external energy fields. The goal may be to generate an immediate failure leading to unauthorized exposure of secure information, or stimulation of premature aging, thereby generating an end of life failure.

T.Lnk_Att - Linked Attacks

An attacker may perform successive attacks with the result that the TOE becomes unstable or some aspect of the security functionality is degraded. A following attack may then be successfully executed.

Monitoring outputs while manipulating inputs in the presence of environmental stress is an example of a linked attack.

T.Rep_Atk - Repetitive Attack

An attacker may utilize repetitive undetected attempts at penetration to expose memory contents or to change security critical elements in the TOE.

Repetitive attempts related to some or all of the other threats discussed herein may be used to iteratively develop an effective penetration of the TOE security. If these attacks can, in all cases, remain undetected, there will be no warning of increased vulnerability.

T.Clon - Cloning

An attacker may clone part or all of a functional TOE to develop further attacks.

The information necessary to successfully clone part or all of a TOE may derive from detailed inspection of the TOE itself or from illicit appropriation of design information.

3.3.2 Threats addressed by the Operating Environment

T.Carrier_Tamper - Chip Modification and Reuse

An attacker may use a modified TOE in an original carrier to masquerade as an original TOE so that information assets can be fraudulently accessed.

Removal, modification, and re-insertion of that TOE into a carrier could be used to pass such a combination as an original. This might then be used to access the assets to be protected.

T.Priv - Abuse by Privileged Users

A careless, willfully negligent, or hostile administrator or other privileged user may create a compromise of the TOE assets through execution of actions which expose the security functions or the protected data.

A privileged user or administrator could directly implement or facilitate attacks based on any of the threats described here.

3.4 Organizational Security Policies

The organizational security policies discussed below are addressed by SCSUG-SCPP compliant TOEs. Each policy is stated in bold type font. It is followed by an application note, in normal font, which supplies additional information and interpretation.

P.Crypt_Std - Cryptographic Standards

Cryptographic entities, data authentication, and approval functions must be in accordance with ISO and associated industry or organizational standards.

Various cryptographic operations such as DES, triple DES, and RSA are well defined. These, or others of similar maturity and definition, should be used for all cryptographic operations in the TOE.

P.Data_Acc - Data Access

Except for a well-defined set of allowed operations, the right to access specific data and data objects is determined on the basis of:

a) the owner of the object

b) the identity of the subject attempting the access

c) the implicit and explicit access rights to the object granted to the subject by the object owner

Once established, conditions for access to data and data objects will never be reduced.

The TOE may be associated with a number of different authorities including the system integrator, card issuer, and system manager. Each of these may have specific rules for accessing the data contained in the TOE. Certain rules can be established in all cases as represented in the access control SFP detailed in security functional requirement FDP_ACF.1. Others need to be explicitly supplied in policy statements determined by the owner of the object in question.

P.File_Acc - File Access

The right to establish files and the access control structure is determined on the basis of:

a) **the owner of the files**

b) **the identity of the subject attempting to perform setup**

c) **the implicit and explicit access rights to the files granted to the subject by the file's owner**

The TOE may be associated with a number of different authorities including the system integrator, card issuer, and system manager. There may be different rules established by each of these regarding the manipulation of files (as distinguished from the data contained therein). Some rules can be established in all cases as described in the information flow control SFP. Others need to be explicitly supplied in policy statements detemined by the owner of the files in question. Note that in the context of this policy, reference to files is interpreted to include other data containers such as objects, databases and other data structures as well as data files.

P.Ident - Identification

The TOE must be capable of being uniquely identified.

The TOE consists of hardware and software elements. The software might be stored in a hard mask (through incorporation in the ROM photomask) or could be stored in non-volatile memory. The hardware could have optional features which might or might not be enabled. An accurate identification must therefore be established for the exact instantiation of the final product compliant to this TOE. This requires unique identification for each TOE.

P.Sec_Com - Secure Communications

Secure communication protocols and procedures shall be supported between the TOE and a trusted terminal.

The TOE may engage in a variety of communications ranging from simple status checking through secure data transfer. At the minimum, the TOE must be capable of establishing a secure channel to a trusted source for application loading or for other potentially privileged commands.

(This page purposely left blank)

4 Security Objectives

4.1 TOE Security Objectives

This section defines the security objectives of the TOE. These security objectives reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. Each objective is stated in bold type font. It is followed by an application note, in normal font, which supplies additional information and interpretation.

O.Audit - Audit

The TOE must provide the means of recording selected security-relevant events, so as to assist an administrator in the detection of potential attacks or misconfiguration of the TOE security features that would leave it susceptible to attack.

Audit capability provides the TOE administrator with the opportunity to review past information on operations to determine if a series of attacks has been mounted against the TOE. Analysis of this type of data can provide early indication of potential system vulnerabilities so that an adequate response can be prepared. Data that might be of interest for inclusion in the audit records could include such things as historical information on chip and operating software detail or on operational information such as activation or deactivation of environmental sensors, failure in checksum calculations, or loading or deleting of applications.

O.Crypt - Cryptography

The TOE must support cryptographic functions in a secure manner.

The TOE must perform any cryptographic operations consistent with established cryptographic usage polices and standards in order to maintain the security level provided by the basic cryptographic functions.

O.D_Read - Data Read Format

The TOE must have a consistent requirement for formatting data passing between modules in the chip.

The TOE must act in a fashion that does not expose information being transferred between processing and storage modules inside the IC to any greater risk of compromise than that derived from long-term storage.

O.DAC - Data Access Control

The TOE must provide its users with the means of controlling and limiting access to the objects and resources they own or are responsible for, on the basis of individual users or identified groups of users, and in accordance with the set of rules defined by the P.Data_Acc Security Policy.

The TOE may have a variety of users, administrators, card issuers, associations, etc., each requiring some control over the assets being handled. Some rules will apply in all cases. These are represented in security functional requirement FDP_ACF.1. The remainder must be explicitly stated as required by the needs of the owners of the data. This objective is distinguished from O.FAC by the specification of data access rules in existing files and file structures.

O.Env_Strs - Environmental Stress

The TOE must protect itself against compromise by having a structure which neither reveals security information nor operates in an insecure fashion when exposed to out of standard conditions (high or low) in the environment, including such factors as temperature, voltage, clock frequency, or external energy fields.

The basic TOE must be designed and fabricated so that it continues to provide security to its critical information, including user assets and internal security information, even when exposed to environmental stress. Environmental stress may be a result of the normal environment in which the TOE is used, but may also be representative of an attack against it. In the event of attack, stress may be the only driving force or it may be used in conjunction with one or several other attacks. This objective should work to prevent disclosure of secure information in any of these conditions.

O.FAC - File Access Control

The TOE must provide its users with the means of controlling and limiting the ability to generate or modify files to the files and resources they own or are responsible for, on the basis of individual users or identified groups of users and in accordance with the set of rules defined by the P.File_Acc Security Policy.

The TOE may have multiple data users and owners needing to maintain security of their own assets in the TOE. General rules for file creation, modification, deletion, and the associated access rights are provided in security functional requirement FDP_IFF.1. Others need to be explicitly provided by the users and owners represented. This objective is distinguished from O.DAC by reference to the file structures themselves and not by the information that may be contained therein. Note that in the context of this objective, reference to files is interpreted to include other data containers such as objects, databases and other data structures as well as data files.

O.Flt_Ins - Fault Insertion

The TOE must be resistant to repeated probing through insertion of erroneous data.

The TOE must prevent the release of information though the analysis of responses to repetitive probing. This objective could also work through the detection of such attacks and the initiation of corrective actions to counter such attempts.

O.I_Leak - Information Leakage

The TOE must provide the means of controlling and limiting the leakage of information in the TOE so that no useful information is revealed over the power, ground, clock, reset, or I/O lines.

The TOE must be designed and programmed so that analysis of such elements as power consumption does not reveal information about processing operations or compromise secure information.

O.Ident - TOE Identification

The TOE must support the recording and preservation of identification information.

The TOE consists of hardware and software elements. The software may be stored in a hard mask (through incorporation in the ROM photomask) or in non-volatile memory. The hardware could have optional features which might or might not be enabled. It is therefore essential that an accurate identification be established for the exact instantiation of the final product compliant to this protection profile. This requires unique identification for each TOE.

O.Init - Initialization

The TOE must assume its initial state immediately upon power-up, reset, or after other restart conditions.

The TOE must always start in a defined and controlled state regardless of how it was reset. This objective works to prevent attacks which attempt to upset the operation and leave the TOE in an undefined state.

O.Life_Cycle - Life-Cycle Functions

The TOE must provide means of controlling and limiting the use of life-cycle-specific commands to the life cycle stages in which they are intended.

The design and implementation of the TOE must be such that the only commands available to a specific operation are related to the ICC life-cycle appropriate to that application. Thus, elements such as debug or one time loading of identification registers should never be available during operational TOE use.

O.Log_Prot - Logical Protection

The TOE must protect itself against logical compromise by having a structure which is resistant to logical manipulation or modification.

The TOE must be designed and programmed so that it resists attempts to compromise its security features through attacks on its logical operation. The TOE must prevent the release of secure information while it is operating properly in the presence of logic probes and command modifications.

O.Mult_App - Multiple Applications

The TOE must support an application (or applications) while providing and maintaining security between and among the various resident elements.

The design and implementation of the TOE must be such that each application or major operational unit can not affect the secure operation of other such applications. This separation must be maintained such that information that is restricted to a single application is not accessible elsewhere, and can not be changed except from within that application.

O.Phys_Prot - Physical Protection

The TOE must be resistant to physical attack or be able to create difficulties in understanding the information derived from such an attack.

The basic TOE must be designed and fabricated so that it requires a combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information, contents of memory, or other information which could contribute to a compromise in TOE security through a physical attack on the IC. This objective should work to prevent disclosure of secure information.

O.Res_Access - Resource Access

The TOE shall protect its resources against monopolization by a user or attacker to the detriment of other users of the TOE.

The TOE should be designed and implemented so that resource allocation is controlled in a manner which supports all intended users.

O.Reuse - Replay

The TOE shall protect its resources against replay attacks.

The TOE must act so that no assets can be compromised through an attacker's attempt to replay or restart an operation which might have been completed successfully or interrupted in process.

O.Search - Data Search

Data and files which are subject to search by unauthorized entities shall be protected from repeated entry.

The TOE may be subject to repetitive read commands in an attempt to recover secure information. It must protect against this.

O.Sec_Com - Secure Communications

The TOE must be able to support secure communication protocols and procedures with a trusted terminal.

The TOE must provide a mechanism for establishing and maintaining a secure infomation link into the CAD.

O.Set_Up - Set-Up Sequence

The TOE must require a defined sequence of operations prior to general utilization.

The TOE must be placed into operation in a controlled and defined manner. This objective acts to prevent use of the TOE before all of the protective measures may be enabled or protective codes entered.

O.Unlink - Linkage

The TOE must allow multiple uses of resources or services without providing any information which, through compilation of many operations, would lead to a compromise in security.

The TOE should be designed and implemented so that no information that would contribute to a breach in security is exposed in any set of normal operations.

4.2 Environment Security Objectives

The following are the protection profile non-IT security objectives that are to be satisfied without imposing technical requirements on the TOE. That is, they do not require the implementation of functions in the TOE hardware and/or software. These security objectives are assumed by the PP to be in place in the TOE environment. They are included as necessary to support the TOE security objectives in addressing the security problem defined in the TOE security environment. Each objective is stated in bold type font. It is followed by an application note, in normal font, which supplies additional information and interpretation.

OE.CAD_Sec-Com - Card Acceptor Device Secure Communication

A trusted CAD is available for secure communication with the TOE.

The CAD is capable of accepting and maintaining a secure communications link with the TOE.

OE.Data_Store - Off-TOE Data Storage

All TOE data stored off of the TOE must be controlled for confidentiality and integrity according to the owner's needs.

A variety of TOE information may be stored separately from the TOE. This may include ownership, issuer data, personalization data, etc. The personnel and systems in charge of this information are responsible for the maintenance of its required security.

OE.Key_Supp - Crypto Key Support

All imported smart card related cryptographic keys must be supported according to the owners' needs.

A variety of keys may be imported for use by, and in conjunction with, the TOE. These may include shared private keys, public/private key pairs, etc. These keys will be supplied from the various bodies controlling the operations of the system in which the TOE is functioning. The personnel and systems in charge of these keys are responsible for the required security of their generation, distribution, maintenance, and destruction.

OE.Perss - Personnel

Personnel working as administrators or in other privileged positions shall be carefully selected and trained for reliability.

Careful selection and training of administrators and others in privileged positions works to detect, prevent, or counter other attacks.

OE.Pwr_Clock - Power and Clock

The CAD supplies power and clock signals to the TOE.

The TOE is internally unpowered, so support must be delivered to the card from the card acceptor device or through an alternate connection to the TOE terminals.

OE.Role_Man - Role Management

Management of roles for the TOE is performed in a secure manner off-card.

The various roles involved in working with the TOE are established in the development and user community through the TOE manufacturers, card issuing bodies, etc. The personnel in charge of these roles are responsible for their management.

OE.Tamper - Tamper Indication

The carrier for the TOE shall provide an indication of tampering if the TOE has been removed and re-inserted.

The personnel in charge of inspecting the TOE carrier are responsible for the detection of tampering with the carrier. This objective can only apply in those cases when the carrier is presented to such personnel and it is physically available for inspection.

(This page purposely left blank)

5 IT Security Requirements

5.1 TOE IT Security Requirements

This section contains the functional requirements that must be satisfied by a SCSUG-SCPP–compliant TOE.

5.1.1 TOE IT Security Functional Requirements

Table 5.1 lists the IT security functional components and indicates whether the component has been refined and if all operations of that requirement are to be met by the TOE. Following the table, each requirement is listed with assignments, selections and refinements indicated in **bold** type. General assignments and selections, requiring definition in the ST are indicated in ***bold italic*** type. These are further delineated in Annex C.2 (Functional Component Operations).

Note that a new Security Functional Component, FAU_LST.1 is referenced. The details of this component and rationale for its inclusion are contained in Section 6.5 (Rationale for Explicitly Stated IT Security Requirements).

Table 5.1 Security Functional Components

Component	Component Name	Refined?	Operations Completed?
FAU_ARP.1	Security alarms	no	no
FAU_LST.1	Audit list generation	Explicitly stated	partial
FAU_SAA.1	Potential violation analysis	no	no
FAU_SEL.1	Selective audit	no	no
FAU_STG.1	Protected audit trail storage	no	yes
FAU_STG.4	Prevention of audit data loss	no	partial
FCS_CKM.1	Cryptographic key generation	no	no
FCS_CKM.3	Cryptographic key access	no	no
FCS_COP.1	Cryptographic operation	no	no
FDP_ACC.1	Subset access control	no	no

Component	Component Name	Refined?	Operations Completed?
FDP_ACF.1	Security attribute based access control	no	no
FDP_ETC.1	Export of user data without security attributes	no	no
FDP_IFC.1	Subset information flow control	no	no
FDP_IFF.1	Simple security attributes	no	no
FDP_ITC.1	Import of user data without security attributes	no	no
FDP_ITT.1	Basic internal transfer protection	no	partial
FDP_RIP.1	Subset residual information protection	no	partial
FDP_UIT.1	Data exchange integrity	no	partial
FIA_AFL.1	Authentication failure handling	no	no
FIA_ATD.1	User attribute definition	no	no
FIA_UAU.1	Timing of authentication	no	no
FIA_UAU.7	Protected authentication feedback	no	yes
FIA_UID.1	Timing of identification	no	no
FMT_MOF.1	Management of security functions behavior	no	partial
FMT_MSA.1	Management of security attributes	no	no
FMT_MSA.2	Secure security attributes	no	N/A
FMT_MSA.3	Static attribute initialization	no	partial
FMT_MTD.1	Management of TSF data	no	no
FMT_MTD.2	Management of limits on TSF data	no	partial
FMT_MTD.3	Secure TSF data	no	N/A
FMT_REV.1	Revocation	no	no
FPT_FLS.1	Failure with preservation of secure state	no	no
FPT_ITI.1	Inter-TSF detection of modification	no	no
FPT_ITT.1	Basic internal TSF data transfer protection	no	yes
FPT_PHP.3	Resistance to physical attack	no	partial
FPT_RCV.3	Automated recovery without undue loss	no	partial
FPT_RCV.4	Function recovery	no	yes

Component	Component Name	Refined?	Operations Completed?
FPT_RPL.1	Replay detection	no	no
FPT_RVM.1	Non-bypassability of the TSP	no	N/A
FPT_SEP.1	TSF domain separation	no	N/A
FPT_TST.1	TSF testing	no	no
FRU_RSA.1	Maximum Quotas	no	no
FTP_ITC.1	Inter-TSF trusted channel	no	no

FAU_ARP.1 - Security alarms

FAU_ARP.1.1 The TSF shall take ***list of the least disruptive actions*** upon detection of a potential security violation.

FAU_LST.1 - Audit list generation

FAU_LST.1.1 The TSF shall be able to generate an audit list of the following auditable events:

a) **production history file**

b) ***other specifically defined auditable events*.**

FAU_LST.1.2 The TSF shall record within each audit record at least the following ***audit relevant information.***

a) **production history file shall contain:**

1. IC type and fabricator

2. IC fabrication date and batch identifier

3. IC serial number

4. Operating software identification and release date

5. IC Module fabricator and packaging date

6. ICC manufacturer and embedding date

7. IC prepersonalization equipment and date

8. ***other specifically defined history events***

b) ***other specifically defined audit relevant information***

FAU_SAA.1 - Potential violation analysis

FAU_SAA.1.1 The TSF shall be able to apply a set of rules in monitoring the audited events and based upon these rules indicate a potential violation of the TSP.

FAU_SAA.1.2 The TSF shall enforce the following rules for monitoring audited events:

a) Accumulation or combination of ***subset of defined auditable events*** known to indicate a potential security violation;

b) ***any other rules***.

FAU_SEL.1 - Selective audit

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) ***object identity, user identity, subject identity, host identity, event type***

b) ***list of additional attributes that audit selectivity is based upon.***

FAU_STG.1 - Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to **prevent** modifications to the audit records.

FAU_STG.4 - Prevention of audit data loss

FAU_STG.4.1 The TSF shall **overwrite the oldest stored audit records** and ***other actions to be taken in case of audit storage failure*** if the audit trail is full.

FCS_CKM.1 - Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified ***cryptographic key generation algorithm*** and specified ***cryptographic key sizes*** that meet the following ***list of standards.***

FCS_CKM.3 - Cryptographic key access

FCS_CKM.3.1 The TSF shall perform ***type of cryptographic key access*** in accordance with a specified ***cryptographic key access method*** that meets the following***: list of standards***.

FCS_COP.1 - Cryptographic operation

FCS_COP.1.1 The TSF shall perform ***list of cryptographic operations*** in accordance with a specified ***cryptographic algorithm*** and ***cryptographic key sizes*** that meet the following: ***list of standards***.

FDP_ACC.1 - Subset access control

FDP_ACC.1.1 The TSF shall enforce the ***access control SFP*** on ***list of subjects, objects, and operations among subjects and objects covered by the SFP***.

FDP_ACF.1 - Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the ***access control SFP*** to objects based on ***security attributes, named groups of security attributes***.

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ***rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects.***

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: ***rules, based on security attributes, that explicitly authorize access of subjects to objects***.

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the rules: ***rules, based on security attributes, that explicitly deny access of subjects to objects.***

FDP_ETC.1 - Export of user data without security attributes

FDP_ETC.1.1 The TSF shall enforce the ***access control SFP and the information flow control SFP*** when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.1.2 The TSF shall export the user data without the user data’s associated security attributes.

FDP_IFC.1 - Subset information flow control

FDP_IFC.1.1 The TSF shall enforce the ***information flow control SFP*** on ***list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP***.

FDP_IFF.1 - Simple security attributes

FDP_IFF.1.1 The TSF shall enforce the ***information flow control SFP*** based on the following types of subject and information security attributes: ***the minimum number and type of security attributes***.

FDP_IFF.1.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: ***for each operation, the security attribute-based relationship that must hold between subject and information security attributes***.

FDP_IFF.1.3 The TSF shall enforce the ***additional information flow control SFP rules***.

FDP_IFF.1.4 The TSF shall provide the following ***list of additional SFP capabilities***.

FDP_IFF.1.5 The TSF shall explicitly authorize an information flow based on the following rules*:* ***rules, based on security attributes, that explicitly authorize information flows***.

FDP_IFF.1.6 The TSF shall explicitly deny an information flow based on the following rules*:* ***rules, based on security attributes, that explicitly deny information flows.***

FDP_ITC.1 - Import of user data without security attributes

FDP_ITC.1.1 The TSF shall enforce the ***access control SFP and the information flow control SFP*** when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2 The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC: ***additional importation control rules***.

FDP_ITT.1 - Basic internal transfer protection

FDP_ITT.1.1 The TSF shall enforce the ***access control SFP and the information flow control SFP*** to prevent the **disclosure or modification** of user data when it is transmitted between physically separated parts of the TOE.

FDP_RIP.1 - Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the **de-allocation of the resource from** the following ***list of objects***.

FDP_UIT.1 - Data exchange integrity

FDP_UIT.1.1 The TSF shall enforce the ***information flow control SFP*** to be able to **transmit and receive** user data in a manner protected from **modification** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether **modification** has occurred.

FIA_AFL.1 - Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when ***number*** unsuccessful authentication attempts occur related to ***list of authentication events***.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall ***list of actions***.

FIA_ATD.1 - User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: ***list of security attributes***.

FIA_UAU.1 - Timing of authentication

FIA_UAU.1.1 The TSF shall allow ***list of TSF mediated actions*** on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

FIA_UAU.7 - Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only **none** to the user while the authentication is in progress.

FIA_UID.1 - Timing of identification

FIA_UID.1.1 The TSF shall allow ***list of TSF-mediated actions*** on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

FMT_MOF.1 - Management of security functions behavior

FMT_MOF.1.1 The TSF shall restrict the ability to **modify the behavior of** the functions **listed below** to ***the authorized identified roles***.

a) **management of *data access levels*, which, once established, shall never be reduced**

b) **management of *actions to be taken* in the event of a security alarm**

c) **maintenance of the violation analysis rules by *adding, modifying, or deleting rules* from the set of rules**

d) **management of *changes to cryptographic key attributes* including key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption)**

e) **management of *actions to be taken in the event of an authentication failure***

f) **managing the *list of actions that can be taken before the user is authenticated***

g) **if an authorized administrator can change the *actions allowed before identification, the managing of the action lists***

h) **managing the *revocation rules***

i) **management of the *list of actions that need to be taken in case of replay***

j) **management of the *conditions under which TSF self testing occurs*, such**

as during initial start-up, at regular interval, or under specified condi tions

k) **management of maximum quotas of resources which may be used**

l) management of additional ***list of functions*** to be detailed in the ST

FMT_MSA.1 - Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the ***access control SFP and the information flow control SFP*** to restrict the ability to ***change_default, query, modify, delete, other operations*** the security attributes ***list of security attributes*** to ***the authorized identified roles***.

FMT_MSA.2 - Secure security attributes

FMT_MSA.2.1 The TSF shall ensure that only secure values are accepted for security attributes.

FMT_MSA.3 - Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce *the* ***access control SFP and the information flow control SFP*** to provide **restrictive** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the ***authorized identified roles*** to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1 - Management of TSF data

FMT_MTD.1.1 The TSF shall restrict the ability to ***change_default, query, modify, delete, clear, other operations*** the ***list of TSF data*** to ***the authorized identified roles***.

FMT_MTD.2 - Management of limits on TSF data

FMT_MTD.2.1 The TSF shall restrict the specification of the limits for **the TSF data listed below** to ***the authorized identified roles***.

a) **management of the *threshold for unsuccessful authentication attempts*;**

b) management of additional ***list of functions*** to be detailed in the ST.

FMT_MTD.2.2 The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: ***actions to be taken***.

FMT_MTD.3 - Secure TSF data

FMT_MTD.3.1 The TSF shall ensure that only secure values are accepted for TSF data.

FMT_REV.1 - Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the ***users, subjects, objects, other additional resources*** within the TSC to ***the authorized identified roles***.

FMT_REV.1.2 The TSF shall enforce the rules ***specification of revocation rules***.

FPT_FLS.1 - Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: ***list of types of failures in the TSF***.

FPT_ITI.1 - Inter-TSF detection of modification

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: ***a defined modification metric***.

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform ***action to be taken*** if modifications are detected.

FPT_ITT.1 - Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from **modification** when it is transmitted between separate parts of the TOE.

FPT_PHP.3 - Resistance to physical attack

FPT_PHP.3.1 The TSF shall resist **environmental stress** to the ***list of TSF devices/elements*** by responding automatically such that the TSP is not violated.

FPT_RCV.3 - Automated recovery without undue loss

FPT_RCV.3.1 When automated recovery from a failure or service discontinuity is not possible, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

FPT_RCV.3.2 For **power failure during operation** the TSF shall ensure the return of the TOE to a secure state using automated procedures.

FPT_RCV.3.3 The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding ***quantification*** for loss of TSF data or objects within the TSC.

FPT_RCV.3.4 The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.

FPT_RCV.4 - Function recovery

FPT_RCV.4.1 The TSF shall ensure that **the security functions involved in rollback and reset functions and the scenario of power loss or TOE withdrawal prior to completion** have the property that the SF either completes successfully, or for the indicated failure scenarios, recovers to a consistent and secure state.

FPT_RPL.1 - Replay detection

FPT_RPL.1.1 The TSF shall detect replay for the following entities: ***list of identified entities***.

FPT_RPL.1.2 The TSF shall perform ***list of specific actions*** when replay is detected.

FPT_RVM.1 - Non-bypassability of the TSP

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

FPT_SEP.1 - TSF domain separation

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The TSF shall enforce separation between the security domains of subjects in the TSC.

FPT_TST.1 - TSF testing

FPT_TST.1.1 The TSF shall run a suite of self tests ***during initial start-up, periodically during normal operation, at the request of the authorized user, and/or at the conditions under which self test should occur*** to demonstrate the correct operation of the TSF.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of TSF data.

FPT_TST.1.3 The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.

FRU_RSA.1 - Maximum quotas

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following ***controlled resource*s** that ***individual user, defined group of users, subjects*** can use ***simultaneously or over a specified period of time***.

FTP_ITC.1 - Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit ***the TSF and/or the remote trusted IT product*** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for ***list of functions for which a trusted channel is required.***

5.1.2 TOE IT Security Assurance Requirements

Table 5.2 lists the IT security assurance components and indicates whether the component has been refined. Following the table, each requirement is listed with refinements identified. These requirements are chosen to be in support of specific objectives or are included consistent with an EAL4 augmented assurance level. Details are presented in Section 6 (Rationale). Augmentation includes AVA_VLA.3 and ADV_INT.1.

Table 5.2 Security Assurance Components

Component	Component Name	Refined?
ACM_AUT.1	Partial CM automation	no
ACM_CAP.4	Generation support and acceptance procedures	no
ACM_SCP.2	Problem tracking CM coverage	no
ADO_DEL.2	Detection of modification	yes
ADO_IGS.1	Installation, generation, and start-up procedures	no
ADV_FSP.2	Fully defined external interfaces	no
ADV_HLD.2	Security enforcing high-level design	no
ADV_IMP.1	Subset of the implementation of the TSF	yes
ADV_INT.1	Modularity	yes
ADV_LLD.1	Descriptive low-level design	no
ADV_RCR.1	Informal correspondence demonstration	no
ADV_SPM.1	Informal TOE security policy model	no
AGD_ADM.1	Administrator guidance	no
AGD_USR.1	User guidance	no
ALC_DVS.1	Identification of security measures	yes
ALC_LCD.1	Developer defined life-cycle model	no
ALC_TAT.1	Well-defined development tools	no
ATE_COV.2	Analysis of coverage	no
ATE_DPT.1	Testing: high-level design	no

Component	Component Name	Refined?
ATE_FUN.1	Functional testing	no
ATE_IND.2	Independent testing - sample	no
AVA_MSU.2	Validation of analysis	no
AVA_SOF.1	Strength of TOE security function evaluation	no
AVA_VLA.3	Moderately resistant	yes

ACM_AUT.1 - Partial CM automation

Developer action elements:

ACM_AUT.1.1D The developer shall use a CM system.

ACM_AUT.1.2D The developer shall provide a CM plan.

Content and presentation of evidence elements:

ACM_AUT.1.1C The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

ACM_AUT.1.2C The CM system shall provide an automated means to support the generation of the TOE.

ACM_AUT.1.3C The CM plan shall describe the automated tools used in the CM system.

ACM_AUT.1.4C The CM plan shall describe how the automated tools are used in the CM system.

Evaluator action elements:

ACM_AUT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_CAP.4 - Generation support and acceptance procedures

Developer action elements:

ACM_CAP.4.1D The developer shall provide a reference for the TOE.

ACM_CAP.4.2D The developer shall use a CM system.

ACM_CAP.4.3D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_CAP.4.1C The reference for the TOE shall be unique to each version of the TOE.

ACM_CAP.4.2C The TOE shall be labeled with its reference.

ACM_CAP.4.3C The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

ACM_CAP.4.4C The configuration list shall describe the configuration items that comprise the TOE.

ACM_CAP.4.5C The CM documentation shall describe the method used to uniquely identify the configuration items.

ACM_CAP.4.6C The CM system shall uniquely identify all configuration items.

ACM_CAP.4.7C The CM plan shall describe how the CM system is used.

ACM_CAP.4.8C The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

ACM_CAP.4.9C The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

ACM_CAP.4.10C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.4.11C The CM system shall support the generation of the TOE.

ACM_CAP.4.12C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

Evaluator action elements:

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ACM_SCP.2 - Problem tracking CM coverage

Developer action elements:

ACM_SCP.2.1D The developer shall provide CM documentation.

Content and presentation of evidence elements:

ACM_SCP.2.1C The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

ACM_SCP.2.2C The CM documentation shall describe how configuration items are tracked by the CM system.

Evaluator action elements:

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_DEL.2 - Detection of modification

Developer action elements:

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

(Refinement) The TOE or parts of it are refined to include at least the following:

a) Design Information

1. **IC specification and technology**
2. **IC design**
3. **IC hardware security mechanisms**
4. **IC software security mechanisms**
5. **photomask**
6. **development tools**
7. **initialization procedures**
8. **access control mechanisms**
9. **authentication systems**
10. **data protection systems**
11. **memory partitioning**
12. **cryptographic programs**

b) Data:

1. **initialization data**
2. **personalization data**
3. **passwords**
4. **cryptographic keys**

c) Test Information

1. **test tools**
2. **test procedures**
3. **test programs**
4. **test results**

d) Physical Instantiations

1. **silicon samples**

2. **bond-out chips**
3. **pre-initialized cards**
4. **pre-personalized cards**
5. **personalized but unissued cards**

ADO_DEL.2.2D The developer shall use the delivery procedures.

Content and presentation of evidence elements:

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

Evaluator action elements:

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1 - Installation, generation, and start-up procedures

Developer action elements:

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

Content and presentation of evidence elements:

ADO_IGS.1.1C The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

Evaluator action elements:

ADO_IGS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

ADV_FSP.2 - Fully defined external interfaces

Developer action elements:

ADV_FSP.2.1D The developer shall provide a functional specification.

Content and presentation of evidence elements:

ADV_FSP.2.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.2.2C The functional specification shall be internally consistent.

ADV_FSP.2.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

ADV_FSP.2.4C The functional specification shall completely represent the TSF.

ADV_FSP.2.5C The functional specification shall include rationale that the TSF is completely represented.

Evaluator action elements:

ADV_FSP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.2.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

ADV_HLD.2 - Security enforcing high-level design

Developer action elements:

ADV_HLD.2.1D The developer shall provide the high-level design of the TSF.

Content and presentation of evidence elements:

ADV_HLD.2.1C The presentation of the high-level design shall be informal.

ADV_HLD.2.2C The high-level design shall be internally consistent.

ADV_HLD.2.3C The high-level design shall describe the structure of the TSF in terms of subsystems.

ADV_HLD.2.4C The high-level design shall describe the security functionality provided by each subsystem of the TSF.

ADV_HLD.2.5C The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

ADV_HLD.2.6C The high-level design shall identify all interfaces to the subsystems of the TSF.

ADV_HLD.2.7C The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

ADV_HLD.2.8C The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_HLD.2.9C The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

Evaluator action elements:

ADV_HLD.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_HLD.2.2E The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_IMP.1 - Subset of the implementation of the TSF

Developer action elements:

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF.

(Refinement) to include at least the following subsets:

a) the subset of the physical structure of the TOE related to:

1. **structure size, organization, and layout**
2. **interconnects and data bus layout**
3. **fuse locations**
4. **physical structure including shielding layers and packaging**
5. **EEPROM manipulation**
6. **RAM access**

b) the subset of the logical structure of the TOE related to:

1. **command range and validity checking**
2. **interrupts and reset function**
3. **secure data checking and manipulation**
4. **availability of commands outside of defined application**
5. **transfer of information between applications or functions**

c) the subset of the structure of the TOE related to unalterability of:

1. serial number and other life-cycle identifiers

2. blocking or elimination of debugging functions

Content and presentation of evidence elements:

ADV_IMP.1.1C The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be internally consistent.

Evaluator action elements:

ADV_IMP.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_IMP.1.2E The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

ADV_INT.1 - Modularity

Developer action elements:

ADV_INT.1.1D The developer shall design and structure the TSF in a modular fashion that avoids unnecessary interactions between the modules of the design.

ADV_INT.1.2D The developer shall provide an architectural description.

Content and presentation of evidence elements:

ADV_INT.1.1C The architectural description shall identify the modules of the TSF.

ADV_INT.1.2C The architectural description shall describe the purpose, interface, parameters, and effects of each module of the TSF.

ADV_INT.1.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.

(Refinement) The description shall particularly address the effective separation of parts of the TOE that are separately developed.

Evaluator action elements:

ADV_INT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_INT.1.2E The evaluator shall determine that both the low-level design and the implementation representation are in compliance with the architectural description.

ADV_LLD.1 - Descriptive low-level design

Developer action elements:

ADV_LLD.1.1D The developer shall provide the low-level design of the TSF.

Content and presentation of evidence elements:

ADV_LLD.1.1C The presentation of the low-level design shall be informal.

ADV_LLD.1.2C The low-level design shall be internally consistent.

ADV_LLD.1.3C The low-level design shall describe the TSF in terms of modules.

ADV_LLD.1.4C The low-level design shall describe the purpose of each module.

ADV_LLD.1.5C The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C The low-level design shall describe how each TSP-enforcing function is provided.

ADV_LLD.1.7C The low-level design shall identify all interfaces to the modules of the TSF.

ADV_LLD.1.8C The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

ADV_LLD.1.9C The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

ADV_LLD.1.10C The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

Evaluator action elements:

ADV_LLD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_LLD.1.2E The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

ADV_RCR.1 - Informal correspondence demonstration

Developer action elements:

ADV_RCR.1.1D The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

Content and presentation of evidence elements:

ADV_RCR.1.1C For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

Evaluator action elements:

ADV_RCR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_SPM.1 - Informal TOE security policy model

Developer action elements:

ADV_SPM.1.1D The developer shall provide a TSP model.

ADV_SPM.1.2D The developer shall demonstrate correspondence between the functional specification and the TSP model.

Content and presentation of evidence elements:

ADV_SPM.1.1C The TSP model shall be informal.

ADV_SPM.1.2C The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

ADV_SPM.1.3C The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

ADV_SPM.1.4C The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

Evaluator action elements:

ADV_SPM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_ADM.1 - Administrator guidance

Developer action elements:

AGD_ADM.1.1D The developer shall provide administrator guidance addressed to system administrative personnel.

Content and presentation of evidence elements:

AGD_ADM.1.1C The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

AGD_ADM.1.2C The administrator guidance shall describe how to administer the TOE in a secure manner.

AGD_ADM.1.3C The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

AGD_ADM.1.4C The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

AGD_ADM.1.5C The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

AGD_ADM.1.6C The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

Evaluator action elements:

AGD_ADM.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_USR.1 - User guidance

Developer action elements:

AGD_USR.1.1D The developer shall provide user guidance.

Content and presentation of evidence elements:

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

AGD_USR.1.3C The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

AGD_USR.1.4C The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

Evaluator action elements:

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1 - Identification of security measures

Developer action elements:

ALC_DVS.1.1D The developer shall produce development security documentation.

Content and presentation of evidence elements:

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

(Refinement) The TOE design and implementation is refined to include at least the following:

a) Design Information

1. **IC specification and technology**
2. **IC design**
3. **IC hardware security mechanisms**
4. **IC software security mechanisms**
5. **photomask**
6. **development tools**
7. **initialization procedures**
8. **access control mechanisms**

9. **authentication systems**
10. **data protection systems**
11. **memory partitioning**
12. **cryptographic programs**

b) Data:

1. **initialization data**
2. **personalization data**
3. **passwords**
4. **cryptographic keys**

c) Test Information

1. **test tools**
2. **test procedures**
3. **test programs**
4. **test results**

d) Physical Instantiations

1. **silicon samples**
2. **bond-out chips**
3. **pre-initialized cards**
4. **pre-personalized cards**
5. **personalized but unissued cards**

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

Evaluator action elements:

ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

ALC_LCD.1 - Developer defined life-cycle model

Developer action elements:

ALC_LCD.1.1D The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2D The developer shall provide life-cycle definition documentation.

Content and presentation of evidence elements:

ALC_LCD.1.1C The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2C The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

Evaluator action elements:

ALC_LCD.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_TAT.1 - Well-defined development tools

Developer action elements:

ALC_TAT.1.1D The developer shall identify the development tools being used for the TOE.

ALC_TAT.1.2D The developer shall document the selected implementation-dependent options of the development tools.

Content and presentation of evidence elements:

ALC_TAT.1.1C All development tools used for implementation shall be well-defined.

ALC_TAT.1.2C The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

ALC_TAT.1.3C The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

Evaluator action elements:

ALC_TAT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_COV.2 - Analysis of coverage

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation of evidence elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

Evaluator action elements:

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_DPT.1 - Testing: high-level design

Developer action elements:

ATE_DPT.1.1D The developer shall provide the analysis of the depth of testing.

Content and presentation of evidence elements:

ATE_DPT.1.1C The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

Evaluator action elements:

ATE_DPT.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_FUN.1 - Functional testing

Developer action elements:

ATE_FUN.1.1D The developer shall test the TSF and document the results.

ATE_FUN.1.2D The developer shall provide test documentation.

Content and presentation of evidence elements:

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

ATE_FUN.1.2C The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

ATE_FUN.1.3C The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.4C The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.5C The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

Evaluator action elements:

ATE_FUN.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2 - Independent testing – sample

Developer action elements:

ATE_IND.2.1D The developer shall provide the TOE for testing.

Content and presentation of evidence elements:

ATE_IND.2.1C The TOE shall be suitable for testing.

ATE_IND.2.2C The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

Evaluator action elements:

ATE_IND.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2E The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

AVA_MSU.2 - Validation of analysis

Developer action elements:

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

Content and presentation of evidence elements:

AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

Evaluator action elements:

AVA_MSU.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

AVA_SOF.1 - Strength of TOE security function evaluation

Developer action elements:

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

Content and presentation of evidence elements:

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

Evaluator action elements:

AVA_SOF.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_SOF.1.2E The evaluator shall confirm that the strength claims are correct.

AVA_VLA.3 - Moderately resistant

Developer action elements:

AVA_VLA.3.1D The developer shall perform and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

AVA_VLA.3.2D The developer shall document the disposition of identified vulnerabilities.

Content and presentation of evidence elements:

AVA_VLA.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

(Refinement) The analysis shall take into account the following generic vulnerabilities:

a) The TOE may be subject to deconstruction to reveal internal circuits and structures.

b) The TOE may be subject to tampering with the structure and content of internal memories, data transport mechanisms, security functions, and test methods.

c) The TOE may be subject to analysis of information which is internal to the device, through monitoring of connections between elements of the circuits and structures.

d) The TOE may be subject to use of logical commands to produce responses that lead to security vulnerabilities.

e) The TOE may be subject to manipulations outside defined operational boundaries that lead to security vulnerabilities.

f) The TOE may be subject to analysis of information that is available external to the device through monitoring emanations or any of the connections to the device including power, ground, clock, i/o, and reset.

g) The TOE may be subject to vulnerabilities that have been identified in preceding generations of the same, or a similar, TOE.

AVA_VLA.3.2C The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

AVA_VLA.3.3C The evidence shall show that the search for vulnerabilities is systematic.

Evaluator action elements:

AVA_VLA.3.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VLA.3.2E The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

AVA_VLA.3.3E The evaluator shall perform an independent vulnerability analysis.

AVA_VLA.3.4E The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

AVA_VLA.3.5E The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a moderate attack potential.

5.2 Security Requirements for the IT Environment

Table 5.3 lists the IT security functional components that apply to the IT environment. No refinements are required.

Table 5.3 Security Requirements for the Environment

Component	Component Name
FCS_CKM.2	Cryptographic key distribution
FCS_CKM.4	Cryptographic key destruction
FMT_SMR.1	Security Roles

FCS_CKM.2 - Cryptographic key distribution

FCS_CKM.2.1 The TSF shall distribute cryptographic keys in accordance with a specified ***cryptographic key distribution method*** that meets the following ***list of standards***.

FCS_CKM.4 - Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified ***cryptographic key destruction method*** that meets the following ***list of standard*s.**

FMT_SMR.1 - Security roles

FMT_SMR.1.1 The TSF shall maintain ***the authorized identified role*s.**

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

6 Rationale

6.1 TOE Description Rationale

The target of evaluation, a smart card, has been defined. This TOE has a unique set of threats relating to its character as a small, self-contained microprocessor that is powered only when connected to a reader, is manufactured in large quantities, and may ultimately be issued to untrusted card holders for their long-term retention. The description of the TOE supports the statement of threats, policies, and assumptions discussed above. It also provides information sufficient to support application notes and the further development of the objectives and requirements.

6.2 Security Objectives Rationale

This section demonstrates that the stated security objectives counter all identified threats, policies, or assumptions.

6.2.1 Security Objectives Coverage

The following tables provide a mapping of security objectives to the environment defined by the threats, policies, and assumptions, illustrating that each security objective covers at least one threat, policy or assumption and that each threat, policy, or assumption is covered by at least one security objective.

Table 6.1 Threats Related to Objectives

Threat	Is Addressed By Objective(s)
T.P_Probe	O.D_Read, O.Phys_Prot
T.P_Modify	O.Phys_Prot
T.E_Manip	O.Phys_Prot
T.Flt_Ins	O.Flt_Ins
T.Forcd_Rst	O.Init
T.Inv_Inp	O.Log_Prot
T.Load_Mal	O.Log_Prot
T.Reuse	O.Reuse
T.Search	O.Log_Prot, O.Search
T.UA_Load	O.Log_Prot
T.Access	O.DAC
T.First_Use	O.Set_Up
T.Impers	O.Set_Up
T.App_Ftn	O.Mult_App
T.LC_Ftn	O.Life_Cycle
T.Res_Con	O.Res_Access
T.Crypt_Atk	O.Crypt
T.I_Leak	O.Env_Strs, O.I_Leak
T.Link	O.Unlink
T.Env_Strs	O.Env_Strs
T.Lnk_Att	O.Audit, O.Env_Strs, O.Flt_Ins, O.Init, O.Life_Cycle, O.Log_Prot, O.Mult_App, O.Res_Access, O.Search
T.Rep_Atk	O.Audit
T.Clon	O.Phys_Prot
T.Carrier_Tamper	OE.Tamper
T.Priv	OE.Perss

Table 6.2 Organizational Security Policies Related to Objectives

Policy	Is Addressed By Objective(s)
P.Crypt_Std	O.Crypt
P.Data_Acc	O.DAC
P.File_Acc	O.FAC
P.Ident	O.Ident
P.Sec_Com	O.Sec_Com

Table 6.3 Assumptions Related to Objectives

Assumption	Is Addressed By Objective(s)
A.CAD_Sec-Com	OE.CAD_Sec-Com
A.Data_Store	OE.Data_Store
A.Key_Supp	OE.Key_Supp
A.Pwr_Clock	OE.Pwr_Clock
A.Role_Man	OE.Role_Man

Table 6.4 Security Objectives Related to Environmental Considerations

Security Objective	Is Necessitated By:
O.Audit	T.Lnk_Att, T.Rep_Atk
O.Crypt	T.Crypt_Atk, P.Crypt_Std
O.D_Read	T.P_Probe
O.DAC	T.Access, P.Data_Acc
O.Env_Strs	T.I_Leak, T.Env_Strs, T.Lnk_Att
O.FAC	P.File_Acc

Security Objective	Is Necessitated By:
O.Flt_Ins	T.Flt_Ins, T.Lnk_Att
O.I_Leak	T.I_Leak
O.Ident	P.Ident
O.Init	T.Forcd_Rst, T.Lnk_Att
O.Life_Cycle	T.LC_Ftn, T.Lnk_Att
O.Log_Prot	T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.Lnk_Att
O.Mult_App	T.App_Ftn, T.Lnk_Att
O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
O.Res_Access	T.Res_Con, T.Lnk_Att
O.Reuse	T.Reuse
O.Search	T.Search, T.Lnk_Att
O.Sec_Com	P.Sec_Com
O.Set_Up	T.First_Use, T.Impers
O.Unlink	T.Link
OE.CAD_Sec-Com	A.CAD_Sec-Com
OE.Data_Store	A.Data_Store
OE.Key_Supp	A.Key_Supp
OE.Perss	T.Priv
OE.Pwr_Clock	A.Pwr_Clock
OE.Role_Man	A.Role_Man
OE.Tamper	T.Carrier_Tamper

6.2.2 Security Objectives Sufficiency

The following discussions provide information regarding:

a) why the identified security objectives provide for effective countermeasures to the threats;

b) why the identified security objectives provide complete coverage of each organizational security policy;

c) why the identified security objectives uphold each assumption.

6.2.2.1 Threats and Objectives Sufficiency

T.P_Probe (Physical Probing of the IC) deals with mechanical attacks on the structure of the TOE itself. It is countered directly by **O.Phys_Prot (Physical Protection)** which ensures that the TOE is constructed using such elements as protective layering, special rules regarding integrated circuit layout, and removal of test pads after initial (wafer) testing is complete. These actions are intended to make deriving information from the IC difficult and, if such information is derived, to make it difficult to interpret and apply such information to attempts to compromise. **O.D_Read (Data Read Format)** ensures that data available on data busses inside the TOE provides no information beyond that which would be available through statically reading the memory.

T.P_Modify (Physical Modification of the IC) deals with attempts to physically modify the TOE such that information relating to the secure operation of the TOE is revealed. This is an extension of T.P_Probe since it may involve physical changes to the IC such as rerouting connections or repairing fuses. This threat is also countered directly by **O.Phys_Prot (Physical Protection)**, which ensures that the TOE is constructed using such elements as protective layering, special rules regarding integrated circuit layout, and removal of test pads after initial (wafer) testing is complete. These actions are intended to make deriving information from the IC difficult and, if such information is derived, to make it difficult to interpret and apply such information to attempts to compromise.

T.E_Manip (Electrical Manipulation of the IC) addresses attempts in which the TOE is modified so that it can be directly fraudulently used. This differs from T.P_Modify in that the goal of the former threat is to derive information and not to reuse the TOE. This threat is also countered directly by **O.Phys_Prot (Physical Protection)**, which ensures that the TOE is constructed using such elements as protective layering, special rules regarding integrated circuit layout, and removal of test pads after initial (wafer) testing is complete. These actions are intended to make deriving information from the IC difficult and, if such information is derived, to make it difficult to interpret and apply such information to attempts to compromise.

T.Flt_Ins (Insertion of Faults) addresses the situation when the TOE is actively being probed through the deliberate insertion of selected inputs with the intent of observing the outputs. This is normally performed over multiple repetitions with small changes in the selected inputs. This is countered through **O.Flt_Ins (Fault Insertion)**, which ensures that such attacks are resisted.

T.Forcd_Rst (Forced Reset) addresses the situations in which the TOE is reset during operation. This may occur at any time including during a reset operation itself. This threat is countered directly by **O.Init (Initialization)**, which ensures that the TOE always enters its defined initial state upon reset.

T.Inv_Inp (Invalid Input) addresses the introduction of input which does not conform to the required style, content, or format. This input may have the look of accidental or erroneous entries (and that may be, in fact, the source of the data) but the result may be the misperformance of the TOE such that security is compromised. Attackers may use non-conforming data, existing but inappropriate commands, or well formatted commands with data requests that refer to locations which are outside of range or not to be utilized in that operation. This threat is countered directly by **O.Log_Prot (Logical Protection)**, which ensures that the TOE is constructed such that it responds in a secure manner to all probing represented by data, commands, or other input which is not fully conforming to the anticipated style and content.

T.Load_Mal (Data Loading Malfunction) addresses the situation in which an attack utilizes maliciously generated errors in the set-up information in an attempt to compromise security. This is related to T.UA_Load except that this threat deals with properly executed loading of corrupted information. This threat is countered directly by **O.Log_Prot (Logical Protection)**, which ensures that the TOE is constructed so that it responds in a secure manner to all probing represented by data, commands, or other input that is not fully conforming to the anticipated style and content.

T.Reuse (Replay Attack) addresses the attempts by an attacker to utilize the information available from a partially or fully completed operation to repeat the operation in a fraudulent fashion. This is countered through **O.Reuse (Replay)**, which ensures that no assets can be compromised in the event of a replay.

T.Search (Data Space Search) addresses the threat of an attacker gaining knowledge of secure information through use of read commands to repetitively search the data space to extract all stored information. This threat is countered by **O.Search (Data Search)**, which prevents repeated entry to data spaces that may be subject to search. This threat is also countered by **O.Log_Prot (Logical Protection)**, which ensures that the TOE is constructed so that it is resistant to logical manipulation.

T.UA_Load (Unauthorized Program Loading) addresses the use of unauthorized programs that either exist in the TOE or are specifically loaded with the intent to penetrate the security features of the TOE. This threat is countered directly by **O.Log_Prot (Logical Protection)**, which ensures that the TOE is constructed such that it responds in a secure manner to all probing represented by data, commands, or other input that is not fully conforming to the anticipated style and content.

T.Access (Invalid Access) addresses the need for protection from unauthorized access to information or resources. This threat is distinguished by the emphasis on access of users to information. This is related to P.Data_Acc and is differentiated from P.File_Acc by the relation to data as opposed to file structures. This threat is countered directly by **O.DAC (Data Access Control)**, which (in conjunction with definitions included in FDP_ACF.1) establishes the access policies.

T.First_Use (Fraud on First Use) deals with fraud perpetrated through the use of a TOE which has not been officially issued. This threat is countered directly by **O.Set_Up (Set-Up Sequence)**, which ensures that a defined and controlled sequence of events is completed before the TOE is enabled for use.

T.Impers (Impersonation) addresses the use of the TOE by an attacker impersonating an authorized user. This threat is countered directly by **O.Set_Up (Set-Up Sequence)**, which ensures that a defined and controlled sequence of events that include appropriate authorizations is completed before the TOE is enabled for use.

T.App_Ftn (Use of Unallowed Application Functions) deals with the exploitation of inappropriate interaction of functions between applications. **O.Mult_App (Multiple Applications)** ensures that such interactions do not compromise security through unauthorized availability of information between applications.

T.LC_Ftn (Use of Unallowed Life Cycle Functions) deals with the exploitation of inappropriate interaction of functions between various life cycle operations. **O.Life_Cycle (Life Cycle Functions)** ensures that such interactions do not compromise security through unauthorized availability of information between elements used in different parts of the life cycle.

T.Res_Con (Resource Contention) addresses the utilization of an excessive amount of memory, program space, or other resource by a negligent user or an attacker, precluding further normal use of the TOE. This threat is countered by **O.Res_Access (Resource Access)**, which ensures that limits on resource allocations are established to preclude this denial of service.

T.Crypt_Atk (Cryptographic Attack) addresses direct attacks on the cryptographic mechanisms employed in the TOE. This threat is countered by **O.Crypt (Cryptography)**, which ensures that any cryptographic functions available are performed in a secure manner.

T.I_Leak (Information Leakage) deals with the exploitation of information inadvertently available from emanations or variations in power consumption or other operating parameters as a function of the operation being performed. SPA and DPA are examples of such information leakage. This threat is countered by **O.I_Leak (Information Leakage)**, which ensures that such information is not exposed. This threat is also partially countered by **O.Env_Strs (Environmental Stress)**, which ensures that the TOE performs in an acceptable fashion (i.e., does not reveal secure information) when exposed to out-of-design-specification conditions.

T.Link (Linkage of Multiple Observations) addresses the observation and linking of a variety of operations, leading to the attacker being able to deduce useful information. This threat is differentiated from T.LC_Ftn and T.App_Ftn since it entails pure observation of normally visible operations and not the manipulation entailed in using operations across defined boundaries. This threat is countered by **O.Unlink (Linkage)** which ensures that information exposed in each operation is of no use to an attacker in understanding and attacking the TOE.

T.Env_Strs (Environmental Stress) deals with the imposition of environmental extremes on the TOE with the intent to cause a direct or indirect failure in the security mechanisms. This threat is countered by **O.Env_Strs (Environmental Stress)**, which ensures that the TOE performs in an acceptable fashion (i.e., does not reveal secure information) when exposed to out-of-design-specification conditions.

T.Lnk_Att (Linked Attacks) deals with multiple attacks synergistically causing a degradation and failure of TOE security. This threat is countered by a number of objectives. **O.Env_Strs (Environmental Stress)** ensures that the TOE performs in an acceptable fashion (i.e., does not reveal secure information) when exposed to out-of-design-specification conditions. **O.Flt_Ins (Fault Insertion)** ensures that active probing of the TOE through the deliberate insertion of selected inputs with the intent of observing the outputs does not result in revealing secure information. **O.Init (Initialization)** ensures that the TOE always enters its defined initial state upon reset. **O.Life_Cycle (Life Cycle Functions)** ensures that exploitation of inappropriate interaction of functions between various life cycle operations does not compromise security through unauthorized availability of information between elements used in different parts of the life cycle. **O.Log_Prot (Logical Protection)** ensures that the TOE remains secure in the event of logical probing attacks. **O.Audit (Audit)** provides the tracking such that multiple attacks may be identified and otherwise countered. **O.Mult_App (Multiple Applications)** ensures that the exploitation of inappropriate interaction of functions between applications does not compromise security through unauthorized availability of information between applications. **O.Res_Access (Resource Access)** ensures that limits on resource allocations are established to preclude denial of service by a negligent user or an attacker **O.Search (Data Search)** prevents repeated entry to data spaces which may be subject to search. With these objectives working together, no information should be revealed, regardless of the stress of multiple attacks on the TOE.

T.Rep_Atk (Audit Failure) represents the implicit threat of non-detection of attacks from other threats. This threat is directly countered by **O.Audit (Audit)**, which ensures that some specified data is recorded and available for analysis so that the nature of repetitive attacks may be detemined and countered.

T.Clon (Cloning) represents the threat that an attacker may manufacture all or a usable portion of the TOE which is then used for fraudulent purposes. This threat is countered by **O.Phys_Prot (Physical Protection)** through a construction that makes it difficult to understand any information derived from physical attacks on the TOE.

T.Carrier_Tamper (Chip Modification and Reuse) deals with the illicit use of a TOE which has been modified and then reinserted in its carrier for fraudulent use. This is a threat in the TOE operating environment since it deals with the TOE carrier. It is differentiated from T.E_Manip and T.Clon, which deal strictly with the TOE. The environmental objective **OE.Tamper (Tamper Indication)** counters this threat by providing for personnel in the environment to provide such detection.

T.Priv (Abuse by Privileged Users) deals with actions by administrators, privileged users and others who may have the capability to compromise the security of the TOE through their actions. This threat is countered by **OE.Perss (Personnel)**, which ensures that these personnel are trained and reliable.

6.2.2.2 Policies and Objectives Sufficiency

P.Crypt_Std (Cryptographic Standards) establishes that accepted cryptographic standards and operations shall be used in the design of the TOE. This is addressed by **O.Crypt (Cryptography)**, which ensures that such standards are used.

P.Data_Acc (Data Access) establishes that there must be a stated policy for access to data and data objects. This is differentiated from P.File_Acc by the relation to data as opposed to file structures. This policy is addressed directly by **O.DAC (Data Access Control)**, which (in conjunction with definitions included in FDP_ACF.1) establishes the access policies.

P.File_Acc (File Access) establishes that there must be a stated policy for the right to establish files and file structures. This is differentiated from P.Data_Acc by the relation to file structures as opposed to data. This policy is addressed directly by **O.FAC (File Access Control)**, which (in conjunction with definitions included in FDP_IFF.1) establishes the access policies.

P.Ident (Identification) establishes that there must be a clear, complete, and unique identification for the TOE. This is addressed through **O.Ident (TOE Identification)**, which ensures that such identification is available.

P.Sec_Com (Secure Communications) establishes that there is a secure communication channel between the TOE and the card acceptor device. This is addressed in **O.Sec_Com (Secure Communications)**, which ensures that the TOE is capable of establishing and using such a link.

6.2.2.3 Assumptions and Objectives Sufficiency

A.CAD_Sec-Com (Card Acceptor Device Secure Communication) establishes that there is assumed to be a secure communication capability in the CAD. This is addressed in **OE.CAD_Sec-Com (Card Acceptor Device Secure Communication)**, which ensures that the CAD is capable of establishing and using such a link.

A.Data_Store (Off-TOE Data Storage) establishes that TOE information, when separate from the TOE, needs to be handled and stored in a secure fashion. **OE.Data_Store (Off-TOE Data Storage)** provides for that security in the environment.

A.Key_Supp (Key Support) establishes that the generation, maintenance, distribution and destruction of keys for proper use of the TOE needs to be supported external to the TOE. **OE.Key_Supp (Crypto Key Support)** provides for that key support in the environment.

A.Pwr_Clock (Power and Clock) establishes that the TOE is internally unpowered and therefore power and clock signals must be delivered from the CAD. **OE.Pwr_Clock (Power and Clock)** provides for that delivery.

A.Role_Man (Role Management) establishes that the roles necessary for proper use of the TOE need to be managed external to the TOE. **OE.Role_Man (Role Management)** provides for that management in the environment.

6.3 Security Requirements Rationale

This section provides the rationale for necessity and sufficiency of security requirements, demonstrating that each of the security objectives is addressed by at least one security requirement, and that every security requirement is directed toward solving at least one objective.

6.3.1 Security Requirements Coverage

The following tables provide a mapping of the relationships of security requirements to objectives, illustrating that each security requirement covers at least one objective and that each objective is covered by at least one security requirement.

Table 6.5 Security Objectives Related to Security Requirements

Security Objective	Is Addressed By:
O.Audit	FAU_LST.1, FAU_SAA.1, FAU_SEL.1, FAU_STG.1, FAU_STG.4
O.Crypt	FCS_CKM.1, FCS_CKM.3, FCS_COP.1
O.D_Read	FDP_ITT.1, FPT_ITT.1, AVA_VLA.3
O.DAC	FDP_ACC.1, FDP_ACF.1, FDP_ETC.1, FDP_IFC.1, FDP_IFF.1, FDP_ITC.1, FIA_ATD.1, FIA_UAU.1, FIA_UID.1, FMT_MOF.1, FMT_MSA.1, FMT_MTD.1, FMT_REV.1
O.Env_Strs	FPT_FLS.1, FPT_PHP.3, AVA_VLA.3
O.FAC	FDP_ACC.1, FDP_ACF.1, FIA_ATD.1, FIA_UAU.1, FIA_UID.1, FMT_MOF.1, FMT_MSA.1, FMT_REV.1
O.Flt_Ins	FDP_ACC.1, FDP_ACF.1, FDP_IFC.1, FDP_IFF.1, FDP_ITC.1, AVA_VLA.3
O.I_Leak	AVA_VLA.3
O.Ident	ACM_CAP.4, ADV_IMP.1
O.Init	FDP_RIP.1, FPT_RCV.3, FPT_RCV.4, FPT_TST.1
O.Life_Cycle	FDP_IFC.1, FDP_IFF.1, FPT_SEP.1, ADV_IMP.1, AVA_VLA.3

Security Objective	Is Addressed By:
O.Log_Prot	FAU_ARP.1, FDP_RIP.1, FIA_UAU.7, FMT_MSA.2, FMT_MTD.2, FMT_MTD.3, FPT_FLS.1, FPT_RCV.3, FPT_RCV.4, FPT_RVM.1, FPT_SEP.1, ADV_IMP.1, AVA_VLA.3
O.Mult_App	FDP_IFC.1, FDP_IFF.1, FPT_SEP.1, ADV_IMP.1, AVA_VLA.3
O.Phys_Prot	ADV_IMP.1, AVA_VLA.3
O.Res_Access	FRU_RSA.1
O.Reuse	FIA_AFL.1, FPT_RPL.1
O.Search	FDP_ACC.1, FDP_ACF.1, FDP_IFC.1, FDP_IFF.1, FIA_AFL.1, FPT_RPL.1, AVA_VLA.3
O.Sec_Com	FDP_ETC.1, FDP_ITC.1, FDP_UIT.1, FPT_ITI.1, FTP_ITC.1
O.Set_Up	FDP_ACC.1, FDP_ACF.1, FIA_UAU.1, FMT_MSA.3, ADV_IMP.1
O.Unlink	FDP_ACC.1, FDP_ACF.1, FDP_ETC.1, FDP_IFC.1, FDP_IFF.1, FTP_ITC.1

Table 6.6 Security Functional Requirements Related to Security Objectives

Security Requirement	Is Necessitated By:
FAU_ARP.1	O.Log_Prot
FAU_LST.1	O.Audit
FAU_SAA.1	O.Audit
FAU_SEL.1	O.Audit
FAU_STG.1	O.Audit
FAU_STG.4	O.Audit
FCS_CKM.1	O.Crypt
FCS_CKM.3	O.Crypt
FCS_COP.1	O.Crypt

Security Requirement	Is Necessitated By:
FDP_ACC.1	O.DAC, O.FAC, O.Flt_Ins, O.Search, O.Set_Up, O.Unlink
FDP_ACF.1	O.DAC, O.FAC, O.Flt_Ins, O.Search, O.Set_Up, O.Unlink
FDP_ETC.1	O.DAC, O.Sec_Com, O.Unlink
FDP_IFC.1	O.DAC, O.Flt_Ins, O.Life_Cycle, O.Mult_App, O.Search, O.Unlink
FDP_IFF.1	O.DAC, O.Flt_Ins, O.Life_Cycle, O.Mult_App, O.Search, O.Unlink
FDP_ITC.1	O.DAC, O.Flt_Ins, O.Sec_Com
FDP_ITT.1	O.D_Read
FDP_RIP.1	O.Init, O.Log_Prot
FDP_UIT.1	O.Sec_Com
FIA_AFL.1	O.Reuse, O.Search
FIA_ATD.1	O.DAC, O.FAC
FIA_UAU.1	O.DAC, O.FAC, O.Set_Up
FIA_UAU.7	O.Log_Prot
FIA_UID.1	O.DAC, O.FAC
FMT_MOF.1	O.DAC, O.FAC
FMT_MSA.1	O.DAC, O.FAC
FMT_MSA.2	O.Log_Prot
FMT_MSA.3	O.Set_Up
FMT_MTD.1	O.DAC
FMT_MTD.2	O.Log_Prot
FMT_MTD.3	O.Log_Prot
FMT_REV.1	O.DAC, O.FAC
FPT_FLS.1	O.Env_Strs, O.Log_Prot,
FPT_ITI.1	O.Sec_Com
FPT_ITT.1	O.D_Read

Security Requirement	Is Necessitated By:
FPT_PHP.3	O.Env_Strs
FPT_RCV.3	O.Init, O.Log_Prot
FPT_RCV.4	O.Init, O.Log_Prot
FPT_RPL.1	O.Reuse, O.Search
FPT_RVM.1	O.Log_Prot
FPT_SEP.1	O.Life_Cycle, O.Log_Prot, O.Mult_App
FPT_TST.1	O.Init
FRU_RSA.1	O.Res_Access
FTP_ITC.1	O.Sec_Com, O.Unlink

Table 6.7 Security Assurance Requirements Related to Security Objectives

Security Requirement	Is Necessitated By:
ACM_AUT.1	selection of EAL4
ACM_CAP.4	O.Ident
ACM_SCP.2	selection of EAL4
ADO_DEL.2	selection of EAL4
ADO_IGS.1	selection of EAL4
ADV_FSP.2	selection of EAL4
ADV_HLD.2	selection of EAL4
ADV_IMP.1	O.Ident, O.Life_Cycle, O.Log_Prot, O.Mult_App, O.Phys_Prot, O.Set_Up
ADV_INT.1	augmentation - see Section 6.8
ADV_LLD.1	selection of EAL4
ADV_RCR.1	selection of EAL4
ADV_SPM.1	selection of EAL4

Security Requirement	Is Necessitated By:
AGD_ADM.1	selection of EAL4
AGD_USR.1	selection of EAL4
ALC_DVS.1	selection of EAL4
ALC_LCD.1	selection of EAL4
ALC_TAT.1	selection of EAL4
ATE_COV.2	selection of EAL4
ATE_DPT.1	selection of EAL4
ATE_FUN.1	selection of EAL4
ATE_IND.2	selection of EAL4
AVA_MSU.2	selection of EAL4
AVA_SOF.1	selection of EAL4
AVA_VLA.3	O.D_Read, O.Env_Strs, O.Flt_Ins, O.I_Leak, O.Life_Cycle, O.Log_Prot, O.Mult_App, O.Phys_Prot, O.Search

6.3.2 Security Requirements Sufficiency

This subsection discusses why the identified SFRs and SARs are sufficient to satisfy the given objective.

O.Audit (Audit) is provided by **FAU_LST.1 (Audit list generation)** for the generation of audit related information. Selection of rules to monitor for potential violations is provided in **FAU_SAA.1 (Potential violation analysis)**. Selection of audit information is provided in **FAU_SEL.1 (Selective audit)**, while protection of the audit data itself is provided in **FAU_STG.1 (Protected audit trail storage)** and **FAU_STG.4 (Prevention of audit data loss)**.

O.Crypt (Cryptography) is provided by **FCS_COP.1 (Cryptographic operation)**. This is supported through **FCS_CKM.1 (Cryptographic key generation)** and **FCS_CKM.3 (Cryptographic key access)** for the generation and validation of the associated secret information.

O.D_Read (Data Read Format) is provided by **FDP_ITT.1 (Basic internal transfer protection)**, which provides the means of preventing the disclosure or modification of user data when it is transmitted between parts of the TOE according to the policies expressed in access control SFP and the Smart Card information flow control SFP. **FPT_ITT.1 (Basic internal TSF data transfer protection)** further specifically protects TSF data from modification. This objective is ensured by **AVA_VLA.3 (Moderately resistant)**, which reviews identified vulnerabilities, including those dealing with probing of the TOE.

O.DAC (Data Access Control) is provided by a combination of requirements. **FDP_ACF.1 (Security attribute based access control)** and **FDP_IFF.1 (Simple security attributes)** set the basic rules through the access control SFP and the information flow control SFP named in **FDP_ACC.1 (Subset access control)** and **FDP_IFC.1 (Subset information flow control). FIA_ATD.1 (User attribute definition)** provides the list of user security attributes. Export and import of user data are controlled through **FDP_ETC.1 (Export of user data without security attributes)** and **FDP_ITC.1 (Import of user data without security attributes)**. The requirements for which actions can be taken prior to imposition of identification are covered in **FIA_UID.1 (Timing of identification)**, while **FIA_UAU.1 (Timing of authentication)** determines when authentication is necessary. **FMT_MOF.1 (Management of security functions behavior)**, **FMT_MSA.1 (Management of security attributes)**, and **FMT_MTD.1 (Management of TSF data)** allow the management of these functions. Finally, **FMT_REV.1 (Revocation)** identifies the roles that are allowed to revoke the security attributes necessary to have access.

O.Env_Strs (Environmental Stress) is provided by **FPT_PHP.3 (Resistance to physical attack)** and **FPT_FLS.1 (Failure with preservation of secure state)**. This objective is ensured by **AVA_VLA.3 (Moderately resistant)**. This requirement provides for the review of identified vulnerabilities, including those dealing with manipulations outside defined operational boundaries.

O.FAC (File Access Control) is provided by a combination of requirements. **FDP_ACF.1 (Security attribute based access)** sets the basic rules through the access control SFP named in **FDP_ACC.1 (Subset access). FIA_ATD.1 (User attribute definition)** provides the list of user security attributes. The requirements for which actions can be taken prior to imposition of identification are covered in **FIA_UID.1 (Timing of identification)**, while **FIA_UAU.1 (Timing of authentication)** covers the requirements for when authentication is necessary. **FMT_MOF.1 (Management of security functions behavior)** and **FMT_MSA.1 (Management of security attributes)** allow the management of these functions. Finally, **FMT_REV.1 (Revocation)** identifies the roles that are allowed to revoke the security attributes necessary to have access.

O.Flt_Ins (Fault Insertion) is provided by the access control SFP and information flow control SFP named in **FDP_ACC.1 (Subset access control)** and **FDP_IFC.1 (Subset information flow control)** and detailed in **FDP_ACF.1 (Security attribute based access control)** and **FDP_IFF.1 (Simple security attributes)**. **FDP_ITC.1 (Import of user data without security attributes)** explicitly addresses the requirements for accepting data. This objective is ensured by **AVA_VLA.3 (Moderately resistant)**, which reviews identified vulnerabilities, including those dealing with logical probing of the TOE.

O.I_Leak (Information Leakage) is provided by **AVA_VLA.3 (Moderately resistant)**. This requirement reviews identified vulnerabilities, including those dealing with the leakage of infomation from the TOE.

O.Ident (TOE Identification) is provided through the assurance requirement **ACM_CAP.4 (Generation support and acceptance procedures)**, which requires the developer to describe and maintain the methods used to uniquely identify the configuration items, which include the TOE. This objective is further supported by **ADV_IMP.1 (Subset of the implementation of the TSF)**, specifically in the implementation of serial number and other life-cycle identifiers.

O.Init (Initialization) is provided through the following requirements. **FDP_RIP.1 (Subset residual information protection)** provides for the protection of information when the resource containing that information is no longer in use. This provides protection to all but the immediately operating elements. **FPT_RCV.3 (Automated recovery without undue loss)**, and **FPT_RCV.4 (Function recovery)** provide for acceptably secure operation in the event of failures. The instance of power failure is of particular concern because of the stated unreliability of supply. Finally, **FPT_TST.1 (TSF testing)**, generates the initial self-test verifying that the TSF is operating correctly.

O.Life_Cycle (Life Cycle Functions) is provided by **FDP_IFF.1 (Simple security attributes)** with the specification of the information flow control SFP named in **FDP_IFC.1 (Subset information flow control)**. **FPT_SEP.1 (TSF Domain separation)** provides the necessary separation and protection to the TSF so that the TSPs can be successfully applied. The implementation of these requirements in the TOE is ensured through **ADV_IMP.1 (Subset of the implementation of the TSF)**, specifically in the implementation of the transfer of information between applications, and by **AVA_VLA.3 (Moderately resistant)** in the review of identified vulnerabilities, including those dealing with manipulations outside defined boundaries and the assurance of secure responses to all logical commands.

O.Log_Prot (Logical Protection) is provided by the requirements discussed below. **FAU_ARP.1 (Security alarms)** provides for a response when selected violations are noted. **FDP_RIP.1 (Subset residual information protection)** provides for the protection of information when the resource containing that information is no longer in use. This provides protection to all but the immediately operating elements. **FMT_MSA.2 (Secure security attributes)** establishes that only secure values can be input for security attributes. **FMT_MTD.3 (Secure TSF data)** provides the same requirements on secure values for TSF data. **FMT_MTD.2 (Management of limits on TSF data)** provides for identifying actions to be taken if limits on TSF data are exceeded. This serves to provide boundaries for potential penetration attempts These requirements work in concert to protect the TOE from penetration by the injection of information into security functions which then might make them insecure. **FIA_UAU.7 (Protected authentication feedback)** provides for the elimination of all feedback during authentication, removing that potential source of information from an attacker. **FPT_FLS.1 (Failure with preservation of secure state)**, **FPT_RCV.3 (Automated recovery without undue loss)**, and **FPT_RCV.4 (Function recovery)** provide for acceptably secure operation in the event of failures. The instance of power failure is of particular concern due to the stated unreliability of supply. **FPT_RVM.1 (Non-bypassability of the TSP)**, along with **FPT_SEP.1 (TSF domain separation)** provide the necessary separation and protection to the TSF so that the required TSPs can be successfully applied. **ADV_IMP.1 (Subset of the implementation of the TSF)** provides for the review and evaluation of the selected subsets of the TOE implementation dealing specifically with resistance to logical attack and **AVA_VLA.3 (Moderately resistant)** reviews identified vulnerabilities, including those dealing with the logical manipulation of the TOE.

O.Mult_App (Multiple Applications) is provided by **FDP_IFF.1 (Simple security attributes)** with the specification of the information flow control SFP named in **FDP_IFC.1 (Subset information flow control)**. **FPT_SEP.1 (TSF Domain separation)** provides the necessary separation and protection to the TSF so that the TSPs can be successfully applied. The implementation of these requirements in the TOE is ensured through **ADV_IMP.1 (Subset of the implementation of the TSF)**, specifically in the implementation of the transfer of information between applications, and by **AVA_VLA.3 (Moderately resistant)** in the review of identified vulnerabilities, including those dealing with manipulations outside defined boundaries and the assurance of secure responses to all logical commands.

O.Phys_Prot (Physical Protection) is provided by the requirements ADV_IMP.1 and AVA_VLA.3. **ADV_IMP.1 (Subset of the implementation of the TSF)** provides for the review and evaluation of the selected subsets of the TOE implementation dealing specifically with resistance to physical attack. **AVA_VLA.3 (Moderately resistant)** provides the review of identified vulnerabilities, including those involving the deconstruction and manipulation of the IC.

O.Res_Access (Resource Access) is provided by **FRU_RSA.1 (Maximum quotas)** which provides for limits on resource allocation.

O.Reuse (Replay) is provided by **FPT_RPL.1 (Replay detection)**. This objective is also supported by **FIA_AFL.1 (Authentication failure handling)** to limit the number of authentication attempts that can be made.

O.Search (Data Search) is provided by the access control SFP and information flow control SFP named in **FDP_ACC.1 (Subset access control)** and **FDP_IFC.1 (Subset information flow control)** and detailed in **FDP_ACF.1 (Security attribute based access control)** and **FDP_IFF.1 (Simple security attributes).** The aspect of replay in searching is provided by **FPT_RPL.1 (Replay detection)**. This objective is also supported by **FIA_AFL.1 (Authentication failure handling)** to limit the number of attempts which can be made. This objective is ensured by **AVA_VLA.3 (Moderately resistant)**. This requirement provides for the review of identified vulnerabilities, including those dealing with logical probing of the TOE.

O.Sec_Com (Secure Communications) is provided by a variety of requirements. **FTP_ITC.1 (Inter-TSF trusted channel)** provides the establishment of a trusted channel. **FDP_ETC.1 (Export of user data without security attributes)** and **FDP_ITC.1 (Import of user data without security attributes)** provide the means of controlling the information which can be exchanged through imposition of the access control SFP and the information flow control SFP. **FDP_UIT.1 (Data exchange integrity)** provides for user data exchange without modification. **FPT_ITI.1 (Inter-TSF detection of modification)** provides the same function for TSF related data.

O.Set_Up (Set-Up Sequence) is provided by the access control SFP named in **FDP_ACC.1 (Subset access control)** and detailed in **FDP_ACF.1 (Security attribute based access control)**. The requirement **FIA_UAU.1 (Timing of authentication)** provides additional support to the generation of the specific set-up sequence. **FMT_MSA.3 (Static attribute initialization)** provides restrictive initial attributes and default values. The implementation of these requirements in the TOE is ensured through **ADV_IMP.1 (Subset of the implementation of the TSF)**, specifically in the implementation of the first time use indicator.

O.Unlink (Linkage) is provided by a combination of requirements. **FDP_ACF.1 (Security attribute based access control)** and **FDP_IFF.1 (Simple security attributes)** set the basic rules for access to data through the access control SFP and the information flow control SFP named in **FDP_ACC.1 (Subset access control)** and **FDP_IFC.1 (Subset information flow control)**. Export of user data is controlled through **FDP_ETC.1 (Export of user data without security attributes)**. **FTP_ITC.1 (Inter-TSF trusted channel)** establishes which functions are involved in any secure transmission. The conjunction of these rules ensures that only that information that is specifically allowed will be exchanged in a fashion that could be observed by an outside entity. This allowed information shall not compromise any security.

6.4 Internal Consistency and Mutual Support

This section demonstrates that the stated security requirements together form a mutually supportive and internally consistent whole. Internal consistency is demonstrated in an analysis of dependencies. Mutual support is shown through consideration of the interactions between and among the SFRs.

6.4.1 Rationale that Dependencies are Satisfied

The selected security requirements include related dependencies, both direct and indirect. The indirect dependencies are those required by the direct dependencies. All of these dependencies must be met or their exclusion justified.

6.4.1.1 Security Functional Requirements Dependencies

The following table provides a summary of the security functional requirements dependency analysis. Justifications for excluded dependencies are in the indicated (following) sections.

Table 6.8 Summary of Security Functional Requirements Dependencies

Component	Depends On:	Which is:
FAU_ARP.1	FAU_SAA.1	included
"	(indirect) FAU_GEN.1	see Section 6.4.1.2
"	(indirect) FPT_STM.1	see Section 6.4.1.3
FAU_LST.1	no dependencies	not applicable
FAU_SAA.1	FAU_GEN.1	see Section 6.4.1.2
"	(indirect) FPT_STM.1	see Section 6.4.1.3
FAU_SEL.1	FAU_GEN.1	see Section 6.4.1.2
"	FMT_MTD.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FPT_STM.1	see Section 6.4.1.3
FAU_STG.1	FAU_GEN.1	see Section 6.4.1.2
"	(indirect) FPT_STM.1	see Section 6.4.1.3
FAU_STG.4	FAU_GEN.1	see Section 6.4.1.2
	(indirect) FPT_STM.1	see Section 6.4.1.3
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	FCS_COP.1 included
"	FCS_CKM.4	see Section 6.4.1.5
"	FMT_MSA.2	included
"	(indirect) FCS_COP.1	included
"	(indirect) FDP_ACC.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFC.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FDP_ITC.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included

Component	Depends On:	Which is:
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
"	(indirect) ADV_SPM.1	included
FCS_CKM.3	FCS_CKM.1 or FDP_ITC.1	both included
"	FCS_CKM.4	see Section 6.4.1.5
"	FMT_MSA.2	included
"	(indirect) FCS_CKM.2	see Section 6.4.1.6
"	(indirect) FCS_COP.1	included
"	(indirect) FDP_ACC.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFC.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
"	(indirect) ADV_SPM.1	included
FCS_COP.1	FCS_CKM.1 or FDP_ITC.1	both included
"	FCS_CKM.4	see Section 6.4.1.5
"	FMT_MSA.2	included
"	(indirect) FCS_CKM.2	see Section 6.4.1.6
"	(indirect) FDP_ACC.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFC.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_MSA.3	included

Component	Depends On:	Which is:
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
"	(indirect) ADV_SPM.1	included
FDP_ACC.1	FDP_ACF.1	included
"	(indirect) FDP_IFC.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_ACF.1	FDP_ACC.1	included
"	FMT_MSA.3	included
"	(indirect) FDP_IFC.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_ETC.1	FDP_ACC.1 or FDP_IFC.1	both included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_IFC.1	FDP_IFF.1	included
"	(indirect) FDP_ACC.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included

Component	Depends On:	Which is:
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_IFF.1	FDP_IFC.1	included
"	FMT_MSA.3	included
"	(indirect) FDP_ACC.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_ITC.1	FDP_ACC.1 or FDP_IFC.1	both included
"	FMT_MSA.3	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	both included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FDP_RIP.1	no dependencies	not applicable
FDP_UIT.1	FDP_ACC.1 or FDP_IFC.1	both included
"	FTP_ITC.1 or FTP_TRP.1	FTP_ITC.1 included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFF.1	included

Component	Depends On:	Which is:
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.1	included
"	(indirect) FMT_MSA.3	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FIA_AFL.1	FIA_UAU.1	included
"	(indirect) FIA_UID.1	included
FIA_ATD.1	no dependencies	not applicable
FIA_UAU.1	FIA_UID.1	included
FIA_UAU.7	FIA_UAU.1	included
"	(indirect) FIA_UID.1	included
FIA_UID.1	no dependencies	not applicable
FMT_MOF.1	FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FIA_UID.1	included
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1	both included
"	FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.3	included
FMT_MSA.2	FDP_ACC.1 or FDP_IFC.1	both included
"	FMT_MSA.1	included
"	FMT_SMR.1	see Section 6.4.1.4
"	ADV_SPM.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_MSA.3	included
FMT_MSA.3	FMT_MSA.1	included

Component	Depends On:	Which is:
"	FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FDP_ACC.1	included
"	(indirect) FDP_ACF.1	included
"	(indirect) FDP_IFC.1	included
"	(indirect) FDP_IFF.1	included
"	(indirect) FIA_UID.1	included
FMT_MTD.1	FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FIA_UID.1	included
FMT_MTD.2	FMT_MTD.1	included
"	FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FIA_UID.1	included
FMT_MTD.3	FMT_MTD.1	included
"	ADV_SPM.1	included
"	(indirect) FIA_UID.1	included
"	(indirect) FMT_SMR.1	see Section 6.4.1.4
FMT_REV.1	FMT_SMR.1	see Section 6.4.1.4
"	(indirect) FIA_UID.1	included
FPT_FLS.1	ADV_SPM.1	included
FPT_ITI.1	no dependencies	not applicable
FPT_ITT.1	no dependencies	not applicable
FPT_PHP.3	no dependencies	not applicable
FPT_RCV.3	ADV_SPM.1	included
"	AGD_ADM.1	included
"	FPT_TST.1	included
"	(indirect) FPT_AMT.1	see Section 6.4.1.7
FPT_RCV.4	ADV_SPM.1	included
FPT_RPL.1	no dependencies	not applicable
FPT_RVM.1	no dependencies	not applicable

Component	Depends On:	Which is:
FPT_SEP.1	no dependencies	not applicable
FPT_TST.1	FPT_AMT.1	see Section 6.4.1.7
FRU_RSA.1	no dependencies	not applicable
FTP_ITC.1	no dependencies	not applicable

6.4.1.2 Justification of Unsupported Dependencies Regarding FAU_GEN.1

Components FAU_SAA.1, FAU_SEL.1, FAU_STG.1, and FAU_STG.4 have direct dependencies on FAU_GEN.1 that are unmet. Component FAU_ARP.1 has an indirect dependency on FAU_GEN.1 which is unmet. As described in Section 6.5 (Rationale for Explicitly stated IT Security Requirements), a new component (FAU_LST.1 Audit list generation), has been defined. This component differs from FAU_GEN.1 only in that it excludes the requirement for date and time in audit records. This is necessitated by the inability of smart card systems (including both the TOE and the environment) to maintain and supply an accurate, reliable date/time reference. The sequence relationships among events to be recorded are all that are routinely available. In all respects, FAU_LST.1 is essentially the same as FAU_GEN.1. Therefore, the dependencies on FAU_GEN.1 are satisfied through FAU_LST.1.

6.4.1.3 Justification of Unsupported Dependencies Regarding FPT_STM.1

Components FAU_ARP.1, FAU_SAA.1, FAU_SEL.1, FAU_STG.1, and FAU_STG.4 have indirect dependencies on FPT_STM.1 that are unmet. As discussed above, however, the sequence relationship of events is all that is required when these functions are associated with the requirements of FAU_LST.1. Therefore, the dependency on FPT_STM.1 is not applicable in this definition.

6.4.1.4 Justification of Unsupported Dependencies Regarding FMT_SMR.1

Components FMT_MOF.1, FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT_MTD.2, and FMT_REV.1 have direct dependencies on FMT_SMR.1 that are unmet. Components FAU_SEL.1, FCS_CKM.1, FCS_CKM.3, FCS_COP.1, FDP_ACC.1, FDP_ACF.1, FDP_ETC.1, FDP_IFC.1, FDP,_IFF.1, FDP_ITC.1, FDP_ITT.1, FDP_UIT.1, and FMT_MTD.3 have indirect dependencies on FMT_SMR.1 that are unmet.

The TOE operates in a role-based mode. It does not maintain lists and controls concerning which individuals belong to which role. This information is maintained in a secure manner off-card as stated in A.Role_Man and OE.Role_Man. Therefore, the dependency on FMT_SMR.1 is not applicable for this TOE.

6.4.1.5 Justification of Unsupported Dependencies Regarding FCS_CKM.4

Components FCS_CKM.1, FCS_CKM.3, and FCS_COP.1 have direct dependencies on FCS_CKM.4 that are unmet.

The TOE, according to A.Key_Supp and OE.Key_Supp, is generally supplied with imported keys (although some locally used keys may be generated in the TOE). All imported keys are assumed to be generated in a secure manner off-card. The keys loaded onto a TOE and the derived keys associated with that single TOE are shared keys, are unique to that TOE, or expire at the termination of that session. Shared keys will appear in many TOEs. Since multiple copies of the TOE may be available for experimentation and probing, the capability to destroy the key in one TOE does not remove the exposure of that key from other TOEs. The capability for destruction of the local and derived keys does not significantly add to the security of the TOE since compromise would only apply to that card and would be useful only during a session which has already been initiated. Further, destruction of keys off the TOE is secure as stated in A.Key_Supp and OE.Key_Supp. Therefore, the dependency on FCS_CKM.4 is not applicable for this TOE.

6.4.1.6 Justification of Unsupported Dependencies Regarding FCS_CKM.2

Components FCS_CKM.3 and FCS_COP.1 have indirect dependencies on FCS_CKM.2 that are unmet.

The TOE is a singular object communicating solely with a card acceptor device. As discussed above in section 6.4.1.5, generation and use of keys are performed in a very limited manner. Such keys are associated with the specific TOE and are not anticipated ever to be transmitted off-card during normal operations. Further, this information is handled in a secure manner off-card as stated in A.Key_Supp and OE.Key_Supp. Therefore, the dependency on FCS_CKM.2 is not applicable for this TOE.

6.4.1.7 Justification of Unsupported Dependencies Regarding FPT_AMT.1

Component FPT_TST.1 has a direct dependency on FPT_AMT.1 that is unmet. Component FPT_RCV.3 has an indirect dependency on FPT_AMT.1 that is unmet.

The TOE depends on the card acceptor device for support of all interactions. The CAD is, however, considered part of the operating environment of the TOE and is not under the control of the TOE itself. It is likely that the CAD may be supplied from a variety of sources, has a variety of uses, and, except under specific conditions represented by A.CAD_Sec-Com and OE.CAD_Sec-Com, cannot be trusted, as stated in A.Pwr_Clock. Also, the entire TSF expressed for this TOE is resident fully on the TOE, with no parts implemented on the CAD. Testing of the abstract machine is therefore not appropriate. The dependency on FPT_AMT.1 is not applicable for this TOE.

6.4.1.8 Security Assurance Requirements Dependencies

The following table provides a summary of the security assurance requirements dependency analysis. All dependencies are satisfied.

Table 6.9 Summary of Security Assurance Requirements Dependencies

Component	Depends On:	Which is:
ACM_AUT.1	ACM_CAP.3	included (hierarchical to ACM_CAP.4)
"	(indirect) ACM_SCP.1	included (hierarchical to ACM_SCP.2)
"	(indirect) ALC_DVS.1	included
ACM_CAP.4	ACM_SCP.1	included (hierarchical to ACM_SCP.2)
"	ALC_DVS.1	included
ACM_SCP.2	ACM_CAP.3	included (hierarchical to ACM_CAP.4)
"	(indirect) ALC_DVS.1	included
ADO_DEL.2	ACM_CAP.3	included (hierarchical to ACM_CAP.4)
"	(indirect) ACM_SCP.1	included (hierarchical to ACM_SCP.2)
"	(indirect) ALC_DVS.1	included
ADO_IGS.1	AGD_ADM.1	included
"	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	(indirect) ADV_RCR.1	included
ADV_FSP.2	ADV_RCR.1	included
ADV_HLD.2	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	ADV_RCR.1	included
ADV_IMP.1	ADV_LLD.1	included
"	ADV_RCR.1	included
"	ALC_TAT.1	included
"	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)

Component	Depends On:	Which is:
"	(indirect) ADV_HLD.2	included
ADV_INT.1	ADV_IMP.1	included
"	ADV_LLD.1	included
"	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	(indirect) ADV_HLD.2	included
"	(indirect) ADV_RCR.1	included
"	(indirect) ALC_TAT.1	included
ADV_LLD.1	ADV_HLD.2	included
"	ADV_RCR.1	included
"	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
ADV_RCR.1	no dependencies	not applicable
ADV_SPM.1	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
	(indirect) ADV_RCR.1	included
AGD_ADM.1	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	(indirect) ADV_RCR.1	included
AGD_USR.1	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	(indirect) ADV_RCR.1	included
ALC_DVS.1	no dependencies	not applicable
ALC_LCD.1	no dependencies	not applicable
ALC_TAT.1	ADV_IMP.1	included
"	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	(indirect) ADV_HLD.2	included
"	(indirect) ADV_LLD.1	included
"	(indirect) ADV_RCR.1	included
ATE_COV.2	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	ATE_FUN.1	included
"	(indirect) ADV_RCR.1	included
ATE_DPT.1	ADV_HLD.1	included (hierarchical to ADV_HLD.2)

Component	Depends On:	Which is:
"	ATE_FUN.1	included
"	(indirect) ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	(indirect) ADV_RCR.1	included
ATE_FUN.1	no dependencies	not applicable
ATE_IND.2	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	AGD_ADM.1	included
"	AGD_USR.1	included
"	ATE_FUN.1	included
"	(indirect) ADV_RCR.1	included
AVA_MSU.2	ADO_IGS.1	included
"	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	AGD_ADM.1	included
"	AGD_USR.1	included
"	(indirect) ADV_RCR.1	included
AVA_SOF.1	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	ADV_HLD.1	included (hierarchical to ADV_HLD.2)
"	(indirect) ADV_RCR.1	included
AVA_VLA.3	ADV_FSP.1	included (hierarchical to ADV_FSP.2)
"	ADV_HLD.2	included
"	ADV_IMP.1	included
"	ADV_LLD.1	included
"	AGD_ADM.1	included
"	AGD_USR.1	included
"	(indirect) ADV_RCR.1	included
"	(indirect) ALC_TAT.1	included

6.4.2 Rationale that Requirements are Mutually Supportive

The requirements represented in this protection profile were developed from a variety of sources including the direct experience of smart card security evaluations by major card associations. As such, the body of requirements has been indirectly shown to be consistent and mutually supportive through its successful application to major commercial systems. A further demonstration is presented below, showing that the security requirements work mutually so that each SFR is protected against bypassing, tampering and deactivation attacks by other SFRs.

In addition to this implicit demonstration of suitability and the details provided for the security functional requirements, the selection of EAL4 with augmentations as explained in a following section provides a consistent and mutually supportive set of assurance requirements.

6.4.2.1 Bypass

Prevention of bypass is derived as described below:

FDP_RIP.1 supports access control and information flow control functions by preventing these SFRs from being bypassed when storage objects are reused and accessed by different subjects.

FIA_UID.1 and FIA_UAU.1 support other functions' allowing user access to data by limiting the actions the user can take prior to identification and authentication.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1 support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FMT_MSA.2, FMT_MSA.3 and FMT_MTD.3 limit the acceptable values for secure data, thus providing protection from bypass to those SFRs dependent on that data.

FPT_PHP.3 provides protection against bypass to all other SFRs by maintaining acceptable security in the event of environmental stress.

FPT_FLS.1, FPT_RCV.3 and FPT_RCV.4 provide for maintenance and recovery of a secure state after failure or service discontinuity, thus preventing bypass of other SFRs.

FPT_RVM.1 prevents bypass of the security functions.

FPT_TST.1 provides for start-up testing to ensure that selected security functions are operational, thus checking for bypass.

6.4.2.2 Tamper

Prevention of tamper is derived as described below:

FAU_LST.1 provides tracking information which may be used to identify tampering with any of the other components.

FAU_STG.1 supports FAU_LST.1 by protecting the integrity of the audit trail.

FAU_ARP.1, with support of FAU_SAA.1, provides for response in the event of detected tampering.

FCS_CKM.1, FCS_CKM.3, and FCS_COP.1 provide for the secure generation and handling of keys, and therefore support those SFRs which may rely on the use of those keys.

FDP_UIT.1 supports FDP_ETC.1 and FDP_ITC.1 by prevention of modification errors during transmission and receipt of user data.

FIA_AFL.1 supports all SFRs dealing with authentication by limiting the number of entry attempts, and then mandating an appropriate action to protect the TOE if too many attempts have been made.

FIA_UID.1 and FIA_UAU.1 support other functions allowing user access to data by limiting the actions the user can take prior to identification and authentication.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1 support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FMT_MSA.2, FMT_MSA.3 and FMT_MTD.3 limit the acceptable values for secure data, thus providing protection from tampering to those SFRs dependent on that data.

FPT_FLS.1, FPT_RCV.3 and FPT_RCV.4 provide for maintenance and recovery of a secure state after failure or service discontinuity, thus preventing tampering with other SFRs.

FPT_PHP.3 provides protection against tampering to all other SFRs by maintaining acceptable security in the event of environmental stress.

FPT_SEP.1 maintains domain separation, and in particular prevents an attacker from tampering with the correct operation of other security functions.

FPT_TST.1 provides for start up testing to ensure that selected security functions are operational, thus checking for tampering.

FRU_RSA.1 maintains limits on resource allocation so no one user or attacker can deny service through monopolization of resources.

6.4.2.3 Deactivation

Prevention of deactivation is derived as described below:

The access control SFP detailed in FDP_ACF.1 and the information flow control SFP detailed in FDP_IFF.1, along with the other SFRs dealing with access control, provide for rigorous control of allowed data manipulations and thus prevent unauthorized deactivation.

The management functions, including FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1, support all other SFRs by restricting the ability to change certain management functions to certain specified roles, thus ensuring that other users cannot circumvent these SFRs.

FMT_MSA.2, FMT_MSA.3 and FMT_MTD.3 limit the acceptable values for secure data, thus providing protection from deactivation to those SFRs dependent on that data.

FPT_FLS.1, FPT_RCV.3 and FPT_RCV.4 provide for maintenance and recovery of a secure state after failure or service discontinuity, thus preventing deactivation of other SFRs.

FPT_PHP.3 provides protection against deactivation to all other SFRs through maintenance of acceptable security in the event of environmental stress.

FPT_TST.1 provides for start up testing to ensure that selected security functions are operational, thus checking for deactivation.

FAU_ARP.1, with support of FAU_SAA.1, provides for response in the event of detected deactivation.

6.5 Rationale for Explicitly stated IT security requirements

A sequence-related audit list function (FAU_LST.1 - Audit list generation) is defined which has the ability to directly specify the audit information to be recorded. This directly supports the security of the TOE while imposing no unnecessary requirements. This function is stated in its entirety as:

FAU_LST.1 - Audit list generation

FAU_LST.1.1 The TSF shall be able to generate an audit list of the following auditable events [assignment: ***specifically defined auditable events*****].**

FAU_LST.1.2 The TSF shall record within each audit record at least the following information [assignment: ***audit relevant information*****].**

This definition is necessitated by consideration of the component FAU_GEN.1 (Audit data generation) which includes the element FAU_GEN.1.2 that states:

> The TSF shall record within each audit record at least the following information: date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event.

The TOE is unpowered except when connected to a reader device. Any time and date information which might be available is dependent on the reader, which is not considered to be a trusted source for this information. Audit data can not, therefore, be linked to time and date but must depend on sequence of operations. Additionally, the memory capacity of the TOE is extremely limited. It is not practical to impose a requirement which introduces overhead not absolutely essential to the security needs of the product. Thus, the audit function in its classical sense is not a useful concept for application to this TOE. At best, the TOE should preserve some information which would be of use in identifying faults and vulnerabilities. This information should include the specification of TOE source, serial number, manufacturer, etc, as indicated in the operations completed for this TOE. Given the limited nature of FAU_LST.1, it is not practical to incorporate the CC defined audit levels of *minimum, basic, detailed,* or *not specified*. It is thus left to the ST and TOE functional specification to provide any further details of audit list generation which may be required to support the intended security.

FAU_LST.1 (Audit List Generation) is modeled on FAU_GEN.1 (Audit Data Generation) which has a dependency on FPT_STM.1 (Reliable Time Stamps). As discussed in this section and in the previous discussion regarding the justification for unmet dependencies on FPT_STM.1, it is not appropriate to include this dependency with FAU_LST.1. There are therefore no dependencies for FAU_LST.1

FAU_GEN.1 (Audit Data Generation) is a dependency for a variety of other requirements. Components FAU_SAA.1, FAU_SEL.1, FAU_STG.1, and FAU_STG.4 have direct dependencies on FAU_GEN.1 and component FAU_ARP.1 has an indirect dependency on FAU_GEN.1. The intent of FAU_LST.1 is identical to that of FAU_GEN.1 in that it requires the generation of some type of audit information which can then be acted upon by the other requirements. FAU_LST.1 is, therefore, an appropriate substitution for FAU_GEN.1 in meeting these dependencies.

6.6 Rationale for Refinement of IT Security Requirements

Five functions have been refined in the definitions contained in this protection profile and are discussed below.

6.6.1 Refinement of ADO_DEL.2.1D

Component ADO_DEL.2.1D has been refined as:

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

(Refinement) The TOE or parts of it are refined to include at least the following:

a) Design Information

1. **IC specification and technology**
2. **IC design**
3. **IC hardware security mechanisms**
4. **IC software security mechanisms**
5. **photomask**
6. **development tools**
7. **initialization procedures**
8. **access control mechanisms**
9. **authentication systems**
10. **data protection systems**
11. **memory partitioning**

12. cryptographic programs

b) Data:

1. initialization data
2. personalization data
3. passwords
4. cryptographic keys

c) Test Information

1. test tools
2. test procedures
3. test programs
4. test results

d) Physical Instantiations

1. silicon samples
2. bond-out chips
3. pre-initialized cards
4. pre-personalized cards
5. personalized but unissued cards

The elements presented in this refinement are those specifically involved with TOE development and fabrication. Each represents information, software, or hardware, the knowledge or possession of which would assist an attacker in defeating the TOE. Including these elements as TOE-specific information that must be addressed is therefore appropriate. Since these refinements provide additional emphasis on the TOE security issues but do not mandate the extension of the procedures or documentation beyond that specified in the basic text of ADO_DEL.2, they do not constitute an extension to the ADO_DEL.2 requirements. Meeting the refined requirement will also meet the original requirement, so this refinement is not an extension of the stated CC requirement.

6.6.2 Refinement of ADV_INT.1.3C

Component ALC_DVS.1.1C has been refined as:

ADV_INT.1.3C The architectural description shall describe how the TSF design provides for largely independent modules that avoid unnecessary interactions.

(Refinement) The description shall particularly address the effective separation of parts of the TOE that are separately developed.

This refinement forces recognition of the fact that the TOE may be a result of the combination of a variety of components, each of which may derive from a different source. As these components are linked together, the possibility of inadvertent or unplanned interactions leading to a lessening of the security must be acknowledged. Thus, the careful review of the interfaces between and among all of these components is warranted. Meeting the refined requirement will also meet the original requirement, so this refinement is not an extension of the stated CC requirement.

6.6.3 Refinement of ALC_DVS.1.1C

Component ALC_DVS.1.1C has been refined as:

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

(Refinement) The TOE design and implementation is refined to include at least the following:

a) Design Information

1. **IC specification and technology**
2. **IC design**
3. **IC hardware security mechanisms**
4. **IC software security mechanisms**
5. **photomask**
6. **development tools**
7. **initialization procedures**
8. **access control mechanisms**
9. **authentication systems**
10. **data protection systems**
11. **memory partitioning**
12. **cryptographic programs**

b) Data:

1. **initialization data**
2. **personalization data**
3. **passwords**
4. **cryptographic keys**

c) Test Information

1. **test tools**
2. **test procedures**
3. **test programs**
4. **test results**

d) Physical Instantiations

1. **silicon samples**
2. **bond-out chips**
3. **pre-initialized cards**
4. **pre-personalized cards**
5. **personalized but unissued cards**

The elements presented in this refinement are those specifically involved with TOE development and fabrication. Each represents information, software, or hardware, the knowledge or possession of which would assist an attacker in defeating the TOE. Including these elements as TOE-specific information that must be addressed is therefore appropriate. Since these refinements provide additional emphasis on the TOE security issues but do not mandate the extension of the procedures or documentation beyond that specified in the basic text of ALC_DVS.1, they do not constitute an extension to the ALC_DVS.1 requirements. Meeting the refined requirement will also meet the original requirement, so this refinement is not an extension of the stated CC requirement.

6.6.4 Refinement of ADV_IMP.1.1D

Component ADV_IMP.1.1D has been refined as:

ADV_IMP.1.1D The developer shall provide the implementation representation for a selected subset of the TSF

(Refinement) to include at least the following subsets:

a) the subset of the physical structure of the TOE related to:

1. **structure size, organization, and layout**
2. **interconnects and data bus layout**
3. **fuse locations**
4. **physical structure including shielding layers and packaging**
5. **EEPROM manipulation**
6. **RAM access**

b) the subset of the logical structure of the TOE related to:

1. **command range and validity checking**
2. **interrupts and reset function**
3. **secure data checking and manipulation**
4. **availability of commands outside of defined application**
5. **transfer of information between applications or functions**

c) the subset of the structure of the TOE related to unalterability of:

1. **serial number and other life-cycle identifiers**
2. **blocking or elimination of debugging functions**

The subsets specifically identified in this refinement are related to one or more threats which the TOE is to counter or policies which the TOE is to implement. The following table links these refinements to their respective objectives, threats, and policies.

Table 6.10 Implementation Subset Refinements

ADV_IMP.1.1D Refinements	Objectives	Threats
ADV_IMP.1.1D (a.1)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
ADV_IMP.1.1D (a.2)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon,
ADV_IMP.1.1D (a.3)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
ADV_IMP.1.1D (a.4)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
ADV_IMP.1.1D (a.5)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
ADV_IMP.1.1D (a.6)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
ADV_IMP.1.1D (b.1)	O.Log_Prot	T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.Lnk_Att
ADV_IMP.1.1D (b.2)	O.Log_Prot	T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.Lnk_Att
ADV_IMP.1.1D (b.3)	O.Log_Prot	T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.Lnk_Att
ADV_IMP.1.1D (b.4)	O.Log_Prot, O.Life_Cycle, O.Mult_App	T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.App_Ftn, T.LC_Ftn, T.Lnk_Att
ADV_IMP.1.1D (b.5)	O.Log_Prot, O.Life_Cycle, O.Mult_App,	T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.App_Ftn, T.LC_Ftn, T.Lnk_Att
ADV_IMP.1.1D (c.1)	O.Ident	P.Ident
ADV_IMP.1.1D (c.2)	O.Life_Cycle	T.LC_Ftn

The subsets specified in this refinement are those involved with specific TOE security related features. It is therefore appropriate to include these subsets in the implementation representation to be reviewed. Since these refinements provide additional emphasis on the TOE security issues but do not mandate the review of the entire implementation, they do not constitute an extension to the ADV_IMP requirements. Meeting the refined requirement will also meet the original requirement, so this refinement is not an extension of the stated CC requirement.

6.6.5 Refinement of AVA_VLA.3.1C

Component AVA_VLA.3.1C has been refined as:

AVA_VLA.3.1C The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

(Refinement) The analysis shall take into account the following generic vulnerabilities:

a) The TOE may be subject to deconstruction to reveal internal circuits and structures.

b) The TOE may be subject to tampering with the structure and content of internal memories, data transport mechanisms, security functions, and test methods.

c) The TOE may be subject to analysis of information which is internal to the device, through monitoring of connections between elements of the circuits and structures.

d) The TOE may be subject to use of logical commands to produce responses that lead to security vulnerabilities.

e) The TOE may be subject to manipulations outside defined operational boundaries that lead to security vulnerabilities.

f) The TOE may be subject to analysis of information that is available external to the device through monitoring emanations or any of the connections to the device including power, ground, clock, i/o, and reset.

g) The TOE may be subject to vulnerabilities that have been identified in preceding generations of the same, or a similar, TOE.

The guidelines presented in this refinement are related to one or more threats which the TOE is to counter. The following table links these refinements to their respective objectives and threats.

Table 6.11 Vulnerability Analysis Refinements

AVA_VLA.3 Refinements	Objectives	Threats
AVA_VLA.3.1C (a)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon
AVA_VLA.3.1C (b)	O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Clon,
AVA_VLA.3.1C (c)	O.D_Read, O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.I_Leak, T.Clon
AVA_VLA.3.1C (d)	O.Flt_Ins, O.Life_Cycle, O.Log_Prot, O.Mult_App, O.Search	T.Flt_Ins, T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.App_Ftn, T.LC_Ftn, T.Lnk_Att
AVA_VLA.3.1C (e)	O.Env_Strs, O.Flt_Ins	T.Flt_Ins, T.I_Leak, T.Env_Strs
AVA_VLA.3.1C (f)	O.I_Leak	T.I_Leak
AVA_VLA.3.1C (g)	O.Log_Prot, O.Phys_Prot	T.P_Probe, T.P_Modify, T.E_Manip, T.Inv_Inp, T.Load_Mal, T.Search, T.UA_Load, T.Lnk_Att, T.Clon

The guidelines presented in this refinement are those involved with specific TOE security related features. Each can be traced to published information on known vulnerabilities. Including these guidelines as TOE-specific information which must be addressed is therefore appropriate. Since these refinements provide additional emphasis on the TOE security issues but do not mandate the extension of the vulnerability analysis beyond previously identified vulnerabilities, they do not constitute an extension to the AVA_VLA.3 requirements. Meeting the refined requirement will also meet the original requirement, so this refinement is not an extension of the stated CC requirement

6.7 Rationale for Strength of Function High

The TOE described in this protection profile is intended to operate in environments that may be under the control of an attacker. Further, the TOE may be exposed to this environment for considerable periods of time (possibly months to years). Since the TOE may represent a significant monetary value, it provides an attractive target which could be attacked repetitively.

Any statistical or probabilistic mechanisms in the TOE may be subjected to prolonged analysis and attack in the normal course of operation. Therefore, such mechanisms should be as resistant to failure as possible, dictating a strength of function-high rating.

A strength of function-high rating is therefore justified on the basis of practicality, cost effectiveness, and efficiency.

6.8 Rationale for Assurance Level EAL4 Augmented

The assurance level for this protection profile is EAL4 augmented.

EAL4 allows a developer to attain a reasonably high assurance level without the need for highly specialized processes and practices. It is considered to be the highest level that could be applied to an existing product line without undue expense and complexity. As such, EAL4 is appropriate for commercial products that can be applied to moderate to high security functions. The TOE described in this protection profile is just such a product.

Augmentation results from the selection of:

AVA_VLA.3 Vulnerability Assessment - Vulnerability Analysis - Moderately resistant

The TOE is intended to function in a variety of applications which may include financial systems. As such, it could contain, represent, or provide access to monetary value. In addition, due to the nature of its intended application, i.e., the TOE may be issued to users and may not be directly under the control of trained and dedicated administrators, the TOE may be subjected to a hostile environment for long periods of time. As a result, it is imperative that the TOE be shown to be moderately resistant to penetration attacks.

EAL4 requires vulnerability assessment through imposition of AVA_VLA.2. This dictates a review of only the identified vulnerabilities. Component AVA_VLA.3 requires, in addition, that a systematic search for vulnerabilities be documented and presented. This provides a significant increase in the consideration of vulnerabilities over that provided by AVA_VLA.2.

The rationale for this augmentation is based on the CEM definitions of basic/medium/high attack potentials. These definitions apply most directly to information processing systems that exist in small numbers and that are offered some form of external protection. The TOE, as discussed above, may be issued in large quantities, is exposed for prolonged periods of time, and is subject to short duration secondary attacks based on longer term development of sophisticated capabilities. As a result, the attack potentials, as stated, are not appropriate. They need to be redefined in this context for the TOE described in this protection profile. With that understanding, a moderate attack potential would address the most reasonably expected competent attacks. Addressing all attacks at all levels (e.g., AVA_VLA.4) introduces cost and complexity higher than justified for all but the most secure applications. It is also questionable if, given the current CEM definitions, this level can be achieved.

AVA_VLA.3 has the following dependencies:

ADV_FSP.1	Informal functional specification
ADV_HLD.2	Security enforcing high-level design
ADV_IMP.1	Subset of the implementation of the TSF
ADV_LLD.1	Descriptive low-level design
AGD_ADM.1	Administrator guidance
AGD_USR.1	User guidance

All of these are met or exceeded in the EAL4 assurance package.

ADV_INT.1 Development - TSF internals - Modularity

The rationale for this augmentation is based on the fact that the TOE is composed of a collection of hardware and software functions ranging from basic operating functions to advanced applications. These may be developed by one or more suppliers. As a result, the operations contained in the final product must have the minimum possibility of destructive interaction. Imposing a requirement on modularity and elimination of unnecessary interactions supports this requirement.

ADV_INT.1 has the following dependencies:

ADV_IMP.1	Subset of the implementation of the TSF
ADV_LLD.1	Descriptive low-level design

All of these are met or exceeded in the EAL4 assurance package

(This page purposely left blank)

Annex A - Glossary

A.1 Common Criteria Terminology

This section contains only those terms which are used in a specialized way in the CC. The majority of terms in the CC are used either according to their accepted dictionary definitions or commonly accepted definitions found in ISO security glossaries or other well-known collections of security terms.

Assets Information or resources to be protected by the countermeasures of a TOE.

Assignment The specification of an identified parameter in a component.

Assurance Ground for confidence that an entity meets its security objectives.

Attack potential The perceived potential for success of an attack, should an attack be launched, expressed in terms of an attacker's expertise, resources and motivation.

Augmentation The addition of one or more assurance component(s) from ISO 15408 Part 3 to an EAL or assurance package.

Authentication data Information used to verify the claimed identity of a user.

Authorized user A user who may, in accordance with the TSP, perform an operation.

Component The smallest selectable set of elements that may be included in a PP, an ST, or a package.

Dependency A relationship between requirements such that the requirement that is depended upon must normally be satisfied for the other requirements to be able to meet their objectives.

Evaluation Assurance Level (EAL) A package consisting of assurance components from ISO 15408 Part 3 that represents a point on the CC predefined assurance scale.

Extension The addition to an ST or PP of functional requirements not contained in Part 2 and/or assurance requirements not contained in ISO 15408 Part 3 of the CC.

Human user Any person who interacts with the TOE.

Identity A representation (e.g. a string) uniquely identifying an authorized user, which can either be the full or abbreviated name of that user or a pseudonym.

Internal communication channel A communication channel between separated parts of TOE.

Internal TOE transfer Communicating data between separated parts of the TOE.

Object An entity within the TSC that contains or receives information and upon which subjects perform operations.

Organizational security policies One or more security rules, procedures, practices, or guidelines imposed by an organization upon its operations.

Package A reusable set of either functional or assurance components (e.g. an EAL), combined together to satisfy a set of identified security objectives.

Protection Profile (PP) An implementation-independent set of security requirements for a category of TOEs that meet specific consumer needs.

Refinement The addition of details to a component.

Role A predefined set of rules establishing the allowed interactions between a user and the TOE.

Secret Information that must be known only to authorized users and/or the TSF in order to enforce a specific SFP.

Security attribute Information associated with subjects, users and/or objects that is used for the enforcement of the TSP.

Security Function (SF) A part or parts of the TOE that have to be relied upon for enforcing a closely related subset of the rules from the TSP.

Security Function Policy (SFP) The security policy enforced by an SF.

Security objective A statement of intent to counter identified threats and/or satisfy identified organization security policies and assumptions.

Security Target (ST) A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.

Selection The specification of one or more items from a list in a component.

Strength of Function (SOF) A qualification of a TOE security function expressing the minimum efforts assumed necessary to defeat its expected security behavior by directly attacking its underlying security mechanisms.

SOF-basic A level of the TOE strength of function where analysis shows that the function provides adequate protection against casual breach of TOE security by attackers possessing a low attack potential.

SOF-medium A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential.

SOF-high A level of the TOE strength of function where analysis shows that the function provides adequate protection against deliberately planned or organized breach of TOE security by attackers possessing a high attack potential.

Subject An entity within the TSC that causes operations to be performed.

Target of Evaluation (TOE) An IT product or system, including its associated administrator and user guidance documentation, that is the subject of an evaluation.

TOE resource Anything useable or consumable in the TOE.

TOE Security Functions (TSF) A set consisting of all hardware, software, and firmware of the TOE that must be relied upon for the correct enforcement of the TSP.

TOE Security Policy (TSP) A set of rules that regulates how assets are managed, protected and distributed within a TOE.

TOE security policy model A structured representation of the security policy to be enforced by the TOE.

Transfers outside TSF control Communicating data to entities not under control of the TSF.

Trusted channel A means by which a TSF and a remote trusted IT product can communicate with the necessary confidence to support the TSP.

Trusted path A means by which a TSF and device physically separated from the TOE can communicate with the necessary confidence to support the TSP.

TSF data Data created by and for the TOE, that might affect the operation of the TOE.

TSF Scope of Control (TSC) The set of interactions that can occur with or within a TOE and are subject to the rules of the TSP.

User Any entity (human user, resident added application, or external IT entity) outside the TOE that interacts with the TOE.

User data Data created by and for the user, that does not affect the operation of the TSF.

A.2 Smart Card Terminology

This section contains only those terms that are used in a specialized way in the smart card industry. The majority of terms are used either according to their accepted dictionary definitions or according to commonly accepted definitions that may be found in ISO security glossaries or other well-known collections of security terms.

Application Intended final use for the smart card. This may include (but is not limited to) such activities as payment, telephony, identification, secure information storage, or loyalty.

Activation A process that gives a card the required operational capability for the cardholder.

Bond-out chips Raw ICs which have been mounted on a small board. Wire bonds are connected from the IC’s input/output pads to the carrier which has contacts on its reverse side. Bond-out chips are sometimes referred to as a module.

Card Acceptor Device (CAD) The mechanism, a key component of reader/writer, into which an integrated circuit (IC) card is inserted.

Card block The IC function related to limiting temporarily the functions allowed to be performed. Card blocking is temporary and can be reset by the proper authorities.

Card disablement The IC function related to terminating all operations other than possibly some limited audit functions. Card disablement is permanent.

Card embedder A manufacturer who assembles a card and integrated circuit.

Card holder A person to whom a card has been legitimately issued (a user).

Card issuer An institution which issues cards to card holders.

Card reader A machine capable of reading and/or writing to a card, such as magnetic stripe card or smart card.

Card Operating System (COS) Operating system developer specific code, written in the microprocessor's native or machine code.

Carrier The holder in which an operational integrated circuit is placed. This is typically the thin, credit card sized piece of plastic that is known as a smart card.

Die The semiconductor IC without any packaging or connections.

Differential power analysis (DPA) A technique combining physical measurements of such things as power consumption with statistical signal processing techniques to identify IC operating details. DPA can, in some instances, provide information leading to recovery of internal operational parameters, keys, etc.

Electrically Erasable Programmable Read Only Memory (EEPROM) A non-volatile memory technology where data can be electrically erased and rewritten.

EMV An integrated circuit card specification for payment systems by Europay, MasterCard, and Visa.

Failure analysis The compilation of techniques used by semiconductor development and testing labs to identify the operating problems in newly designed or modified integrated circuits. Such techniques include not only observation (to determine what is not functioning properly) but also modification of IC internal structure (to determine fixes).

First use indication The IC function related to setting a specific audit bit indicating that the smart card is now in the issued, operational state and can be used for its intended function.

GSM Global system for mobile communication.

Integrated Circuit (IC)
Electronic component(s) contained on a single chip and designed to perform processing and/or memory functions.

Integrated Circuit Card (ICC)
A card into which has been inserted one or more ICs.

Initialization
The process of writing specific information into Non-Volatile Memory during IC manufacturing and testing as well as executing security protection procedures by the IC manufacturer.

Life cycle identifiers
The specific identification of chip fabricator identifier, operating software identifier, chip module identifier, chip embedder identifier, initializer identifier, initialization equipment identifier, personalizer identifier, and personalization equipment identifier.

Modules
A functional assembly for use with other assemblies. These may be separate parts of an IC (CPU, Coprocessor, ROM, RAM, etc.), bond-out chips, or software components.

Non-volatile memory
A semiconductor memory that retains its content when power is removed. (i. e. ROM, EEPROM, FLASH).

Operational keys
The cryptographic keys loaded into the assembled smart card product for use by the cardholder during normal operation.

Operating Software (OS)
That software resident on the TOE which is required for TOE operation up to supporting secure load. This may, or may not, be a full operating system in the conventional sense.

Personalization
The process of writing specific information into the non-volatile memory in preparing the IC for issuance to users.

Photomask
A mask which is used during chip manufacturing to protect selected parts of a silicon wafer from a light source while allowing other parts of the surface of the wafer to be exposed. The purpose is to expose the photoresist on the surface so that subsequent etching processes can generate the desired substrate structure. The photomask is the means by which the chip's circuits, and therefore its functionality, are placed on the chip.

Pilot
A test application in which a system is deployed to a limited geographic area or card holder population so that data on acceptability and operational capability can be gathered prior to full-scale introduction.

Platform A term representing an operational smart card system.

Post-issuance The time period during which the smart card is in the hands of the card holder. In some smart cards, additional functionality can be loaded into the smart card post-issuance.

Production keys The cryptographic keys loaded into the IC for security during production.

Random Access Memory (RAM) A volatile, randomly accessible memory (used in the IC) that requires power to maintain data.

Read Only Memory (ROM) A non-volatile memory (used in the IC) that requires no power to maintain. ROM data is often contained in one of the numerous masks used during manufacture.

Reverse engineering The compilation of techniques used by semiconductor development and testing labs to generate design documentation and specifications for an unknown integrated circuit. Reverse engineering, in its most complete sense, would allow the identification of a complete fabrication package given only an unidentified integrated circuit as a starting point.

Subscriber Identification Module (SIM) A smart card having a shape in accordance with ISO 7812 (ID 0), designed to be inserted into a special cavity in a mobile phone, (necessary for the operation of a GSM phone)

Simple Power Analysis (SPA) A technique in which physical measurements of power consumption over time are used to identify IC operating details. SPA can, in some instances, provide information leading to recovery of internal operational parameters, keys, etc.

Smart card A shaped piece of plastic or other carrier with a small computer chip embedded into it.

Terminal The device used in conjunction with the CAD at the point of transaction.

Transport keys The cryptographic keys loaded into the IC for security during transport of ICs, modules, and assembled products prior to issuance.

(This page purposely left blank)

Annex B Smart Card Technology

Smart cards contain a fully functioning computer system built on a single chip. This computer system has important similarities to and differences from other kinds of computers. Like others, it has a central processing unit (CPU) and various kinds of memory. Unlike others, cost is a major constraint, as the final chips must be sold for a few dollars rather than the tens to hundreds of dollars other computer chips sell for. The chips must also be as small as possible for both cost and reliability reasons.

The Common Criteria was written against a background of traditional information technology, which generally utilizes computers that are typically larger and potentially more networked than those encountered in smart cards. Smart card technology background is therefore highly useful for understanding the security requirements of smart cards. The primary features that impact on security and the Common Criteria are discussed in the following sections.

B.1 Unique Features

B.1.1 Types of Memory

Smart card chips use several types of memory, all implemented on a single chip. These are permanent memory, programmable nonvolatile memory, and volatile memory. Permanent memory is generally ROM (Read Only Memory), which is put in the chip hardware when it is manufactured. It can not be changed, although its operation can be logically blocked. Programmable nonvolatile memory is generally EEPROM (Electrically Erasable Programmable Read Only Memory). It can be programmed after chip manufacture, which is both its strength and its weakness. Its use permits making changes to programs, thus increasing its flexibility but also exposing it to various types of attacks. Volatile memory is generally RAM (Random Access Memory), used as a temporary storage area for interim operations. It loses its contents when power is removed from the chip.

Smart card programming requires special skills, as programs must be written using the smallest amount of memory possible. There is insufficient room for virus checking software, but also not much room for viruses.

A major factor in understanding smart cards is realizing the implications of when a program is added to a specific type of memory. If added in ROM, it must be added when the basic chip is manufactured, and no changes can be made to it. If added in EEPROM, it can be added prior to issuance of the card to the card holder, or afterwards.

There are also new types of memory on the horizon, including FLASH and others. As these are not yet being deployed in significant quantities in financial applications, they are not discussed further here.

B.1.2 Memory and Processing Power

Most smart cards in 2000 use 8-bit microprocessors. Although more powerful 16- and even 32-bit chips will be available shortly, none, as yet, have multi-threading and other powerful features that are common in standard computers.

Memory sizes range from as little as 1K of programmable non-volatile memory to as much as 24K, with larger memory chips coming soon. ROM size is similarly limited. However large memory may become, the total amount will always be relatively limited compared to normal computer capabilities. That limitation imposes the requirement of strict discipline in coding, and limits the defensive measures that can be implemented.

B.1.3 Chip Families

A chip family has a single CPU with many different memory configurations. This is to accommodate programs of varying sizes and states of maturity. The smallest chip in the family may have 4K of ROM, 256 bits of RAM, and 1K of EEPROM. The next size up may have the same ROM and RAM, but varying amounts of EEPROM - 2K or 4K, for example. The next members of the family may have 6K of ROM, 256 bits of RAM, and 6, 8, or 10K of EEPROM. Another possibility is 8K ROM and 4 or 6K of EEPROM. A crypto coprocessor may also be added for those applications that require faster execution of cryptographic algorithms, sometimes with some additional dedicated RAM.

B.1.4 Soft Mask or Hard Mask

It is common in the smart card industry to speak of a card as being either soft mask or hard mask, referring to where an application is placed. If the application is placed in EEPROM, it is termed a soft mask card. If most of the application is placed in ROM, it is called a hard mask card, though variable features and personalization data will still be placed in EEPROM.

It is common practice to use a soft mask card for pilots and then to move on to hard mask cards for larger deployments. However, some applications have limited deployments that are never taken to hard mask, as hard masking is expensive in both time and money. Hard masks may also not be justified for some applications, such as an employee identification card for a small firm.

This differentiation works well for single application cards, but may become confusing if there are multiple applications. That is, a single card may have one application in ROM and thus be considered a hard mask card with respect to that application, but also have another application placed in EEPROM, and so be a soft mask card with respect to that application. As a further example, a particular multiple application card may have a national or international financial application in ROM (with options and personalization data in EEPROM), and any of several loyalty applications in EEPROM. This uses the advantages of each technology for the different applications.

B.1.5 Programming Languages

Most smart cards are currently programmed in low level languages based on proprietary smart card operating software. Some of the programming has been done in the chip's native instruction set (generally Motorola 6805, Intel 8051, or Hitachi H8). This results in highly efficient code, which is much more difficult to program than higher level languages. The number of programmers who can do this programming has been limited, which in itself provides some degree of security.

A new type of card, is coming to market, termed a reconfigurable card. These reconfigurable smart cards have a more robust operating software, which permits adding or deleting application code after the card is issued. Such cards are generally programmed in Java^{TM}, Windows for Smart Cards^{TM}, or MEL^{TM} (the MULTOS^{TM} programming language). These cards may have a card operating software and additional layers that offer industry or application specific features. The operating software must ensure that only authorized applications can be added after the card is issued to the cardholder and that deletions of applications are only done under proper authorization. These reconfigurable cards use programming languages that are very well known in the software community, which is one of their advantages. Many programmers will be able to write smart card programs able to run on this operating software, although the special skill of being able to write very memory efficient programs will still be needed. The security earlier provided by the small number of programmers who knew the proprietary languages will be reduced by this new operating software. Other security features are required to offset the consequent vulnerability.

B.1.6 Off-Line Operation

Smart cards may be used in on-line as well as off-line operation. That is one of their significant advantages. Therefore, countermeasures which depend on network monitoring alone or assume solely off-line operation will be generally ineffective during at least part of the operation of the smart card. This mixture of operation must be carefully considered in the development and specification of countermeasures.

B.1.7 Possession

Smart cards are in the possession of the cardholder all the time. The cardholder may be motivated to fraudulently change some of the data on the card (e.g., balance on a stored-value card, age on an identification card, etc.). An attacker may be attacking his own card or may steal one or several. The attacker can take the card to a well-equipped lab and subject it to all sorts of attacks. This type of attack is far less likely with the more familiar information technology product.

B.1.8 Physical Attacks

Generally, logical attacks can be evaluated separately from physical attacks. This can be done with smart cards to some extent, but not entirely. Physical attacks utilizing techniques derived from semiconductor engineering must be evaluated or the evaluation effort is inadequate. Just as there is a unique synergy in the way that a smart card uses hardware and software to accomplish its tasks, such a combination can also be used by attackers. Hardware-based defenses that might be effective if used properly can be breached by software that does not utilize those defenses to best advantage.

B.1.9 Applications

An application is a program that does something a user wants to have done. Typical applications for smart cards include:

- Financial – Payment schemes may include credit, debit, stored value purse, stored token, or mass transit (generally dedicated to a single transport system and typically having low value).
- Telephony – The primary use is the Subscriber Identification Module (SIM) for digital mobile telephones.
- Identification – Various public and private schemes provide identification credentials to participants. These may be government, corporate, university, or other entities. The identification credentials are typically associated with various rights and duties, defined by the identification provider. These can include membership, driver's licenses, benefit access, passports, national identification, etc. Typically the identification credentials have value in great part because they can not be easily altered by the credential holder; thus assets in the credential must be protected against alteration by the cardholder. Digital certificates used in public key systems fit into this category.
- Secure information storage – Information that is usefully stored in a secure fashion includes health records, health insurance, and other medical type information.
- Loyalty – These are programs like the frequent flyer point programs used by airlines. Points are added and deleted from the card memory in accordance with program rules. The total value of these points may be quite high and it must be protected against improper alteration in much the same way that currency value is protected.
- Networked applications – Smart cards can hold access credentials such as passwords that authenticate a user to a computer network.

These applications may range from very simple to very complex. For example, a loyalty application may be no more than an identifying code, such as a hotel or airline frequent user account code. Most of the information (preferences, total points, etc.) is stored on a mainframe computer somewhere; the card is only used to access the account accurately. The application becomes more complex as more of the information and processing is moved from the computer(s) on the network to the card. Payment applications are typically quite complex, as one of the main reasons for moving from magnetic stripe to smart cards is to permit off-line transactions to be made more securely.

An application may consist of a core of mandatory features and additional variable options that some, but not all, users of the application desire. A loyalty program may be offered with an optional PIN, for example. PIN processing takes up chip real estate and thus increases the cost of the chip, but may be desired to protect high value assets. One user may decide that only lower value assets will be protected by the card and chose the cheaper, non-PIN version. Another makes the opposite decision and chooses the more expensive PIN option. In order to accommodate both, the manufacturer may put all the mandatory features in ROM and the variable options in EEPROM. Common options include personalization data, which will be different for each individual card user. This data is always put in EEPROM, as the cost of putting it in ROM would be prohibitive.

Each of these applications may have different security requirements, security features, roles, and environmental considerations (e.g., whether always used on-line, always used off-line, usually off-line with the capability of going on-line, etc.). The security requirements for the operating software, applications, and the procedures for adding or deleting those applications must therefore be clearly identified and the security functions that are present must be appropriate to the type and intended use of the card.

B.1.10 Cost and Availability

Most of the products envisaged by the Common Criteria have a significant cost (hundreds to thousands of dollars) and are somewhat limited in availability. Smart cards, however, range in cost from a few dollars to several tens of dollars (US). This means that attackers can be expected to be able to buy, or otherwise acquire, multiple copies of the TOE with which to experiment. Destroying some of them in the course of exploration may be considered normal practice.

Most successful smart card projects anticipate issuing hundreds of thousands if not millions of the same card. This has critically important security implications.

- Attackers should be assumed to be able to get multiple copies of cards.
- The asset protected by a single card may be low in value, but the total assets protected by the total card base may be very large.
- The cost of attacking a single card may not be cost effective, but if that successful attack makes subsequent attacks on similar cards easy, the aggregate benefit may justify the investment.
- Initial attacks may require expensive reverse engineering of the smart card, after which subsequent attacks may be much easier and faster.

These conditions are not simply a matter of listing a new threat; they require rethinking all threats in terms of probability and ease.

B.1.11 Cost Sensitivity

The market for smart cards is highly cost sensitive; differences of a few cents per card matter when millions of units are involved. This means that any defensive measures must meet very stringent cost effectiveness tests that are unusual with other IT products.

B.2 Life Cycles

Smart cards are a product composed of physical elements such as the basic integrated circuit, wiring, bond pads, connection pads or antennas, etc., as well as software. The software may be physically incorporated in the integrated circuit through hard masks and circuit modifications, or could be added to programmable non-volatile memory at many different points during the life of the smart card itself.

In addition to the actual instantiation of the smart card product, there are many uses for smart cards and the chips that are used in them. The security needs of smart cards and smart card chips, range from nonexistent to high. Security generally has a cost in money, time to market, and chip real estate, and smart card markets are very price sensitive. Building a smart card typically involves a constant concern for cost containment and often means a trade-off of cost against other desirable things. Various decisions are possible and are often driven by conflicting and quickly changing technological, security, and market needs. The same application can be instantiated in a relatively low security chip for a limited pilot deployment and then in a higher security chip for a larger scale deployment. Some national markets may require a medium level of security, while others demand and are willing to pay for a higher level.

As a result the smart card life cycle can be very complex, involving multiple developers, progressing over several years, with many paths, and following an evolutionary growth. This annex addresses the smart card life cycle in order to clarify some of the understandings and interpretations required to understand and apply the SCSUG-SCPP and to adapt Common Criteria security elements to the smart card product.

This discussion focuses on smart cards used for banking applications, but is not necessarily restricted to those applications. There may be similar needs and life cycles in health care, identity, and a variety of private and public sector areas. It is left up to those market segments to determine whether their needs are the same or different. This discussion may help clarify what the differences are where they exist.

It must be noted that the life cycle discussed here represents the process of developing silicon, software, and systems to perform useful functions. This is a somewhat different description from the design-process life cycle referred to in ALC_LCD (Life Cycle Definition). This latter life cycle deals with the design process of a single product, identifying more of the elements of how to manage a design, when and how to review, change tracking, approvals, etc. It thus deals more with the process and less with the instantiation of the product itself. Such a design-process life cycle will, of necessity, be followed in each stage of the smart card production life cycle but may be different for each developer or fabricator. A smart card, however, is usually a composite product. There may be separate life cycles for the chip, the card, and the application. The same chip can be used for many applications, and the same application can be instantiated on many different chips. The details of the design-process life cycle of a specific smart card as specified in ALC_LCD are left to the security target.

There are potentially many steps in a smart card life cycle and many options in terms of who exe cutes these steps. The Common Criteria roles of “Developer” and “User” need to be refined to reflect the complexities of the situation. The developer roles include the development of the integrated circuit, the operating system, the application software, and the card, with its printing and (optionally) magnetic stripe, bar codes, holograms, or other features. Typically the integrated circuit developer is separate from the card developer, who may, or may not, be the same entity as the operating system and application software developer. The “User” in a banking application is a financial institution that acts as the issuer to the consumer who will use it in financial transactions with a merchant (seller of goods and/or services). There may be a card issuer separate from the application issuer, and there may be several application issuers in the same multiple application card.

There is no single typical smart card life cycle. There are several cycles and several routes through them, including four distinct life cycles based on type of card. These include soft mask, hard mask, proprietary, and reconfigurable. The general elements of the smart card life cycle are described below followed by a detailed section for soft mask, hard mask, proprietary, and reconfigurable cards. There is a separate life cycle for applications, which may intersect with several card life cycles and which drives them. The discussion below deals with these as ideal types, and is most relevant to single application cards. Multiple application cards are more complex, as noted in a following section.

B.2.1 General Life Cycle Model

Minimally, a smart card must meet some user requirements, without which there will be no market for the final product. It must be designed, manufactured, issued, used, and taken out of use. Each of these steps requires discussion.

B.2.1.1 General Model User Requirements

All smart cards begin with user requirements. These requirements drive the rest of the process. Among the issues that may be noted in the user requirements are the following:

- whether there should be one or more than one application
- the level(s) of security needed
- whether the requirements are clearly or only partially known
- whether one or more than one suppliers are desired
- what level and kind of information technology skill the potential users are expected to have

- how much flexibility the product should afford the user
- whether the use is expected to always be on-line to a network or whether some or all of it is expected to function off-line
- legacy issues - whether a new system is being designed or whether the product is expected to be integrated into an existing system
- regulatory issues
- acceptable price ranges
- whether the card will be used in a relatively closed or a relatively open system
- how many cards are anticipated

The user requirements may be well known and have high security needs such as those for the secure application module (SAM) implemented on a smart card in some stored value systems. In such cases a proprietary card is called for, with a specially designed application that merges the application and operating system to create the program code, eliminating any distinction between them. The program code then dictates the chip requirements for this card.

Alternatively, the user requirements may initially be generally stated and only partially known, both to the users and the developers involved. They may also be incomplete, in the sense that additional details must be specified in order to build a functioning card.. In the case of the EMV specifications, there were three different international organizations involved, each of whom not only wanted scope to tailor the application to specific regional, national, and market needs, but also wanted international interoperability. This more complicated kind of application typically goes through a series of different cards, refining the user requirements as time progresses.

B.2.1.2 General Model Design

Smart cards put applications and operating systems into integrated circuits that are embedded in a printed card. Each of these (application, operating system, integrated circuit, and card) is generally designed and manufactured by a different company, although the application and the operating system are sometimes done by the card company.

The design stage may occur sequentially or simultaneously. In a proprietary card, the application design dictates the program code, which dictates the chip design. The smallest possible size of chip will be designed, both for reliability reasons and to preclude the addition of unauthorized additional code. Security features (e.g., environmental sensors, physical structure, etc.) will be included in the chip design. The application also dictates the card design (full size or SIM size, printed with artwork or just inventory control information, etc.). Proprietary cards are time consuming and expensive, as all the cost of the work will have to be amortized over relatively few cards.

Consequently, operating system, chip and card designers frequently try to anticipate several users' requirements and design general purpose operating systems and chips. If the application has lower security requirements or anticipates a small number of cards, or if all the details of the successful application are not yet known, a decision may be made to use one of these general purpose products.

Whether the card is a proprietary one or not, there will be several designers, often working for different companies, working on the application, operating system, program code, chip, and card designs. On a project designing a proprietary card these may form a single, well managed team. If the application designers choose general purpose operating systems and chips, they will have to take into account the fact that these were not designed with clear knowledge of each other's needs.

B.2.1.3 General Model Manufacturing

Manufacturing involves several developers; application, operating system, chip, and card manufacturing may all be done in separate companies. All must have appropriate security arrangements, but these are different depending on the kind of product and company involved.

Integrated circuits are tested after the wafer is manufactured and at several other points during development. One potential attack is to place the chip in to test mode; this should not be possible after the chip has passed beyond the test phase of the life cycle.

Application(s) may be added during the chip manufacturing stage, when the card is manufactured, or after the card has been issued. The chip manufacturer may not ever know what application(s) is being added to his chip if the application is added during card manufacture or after the card has been issued.

B.2.1.4 General Model Issuance

The application issuer may or may not also be the card issuer. The card issuer may authorize other applications to be placed on the card, each with their own life cycles and requirements. The card may or may not be personalized to an individual, depending on the application's requirements. A banking application is issued by a financial institution, which has a contract with the end user that governs use of the application. Debit and credit applications typically are used to access an account at the issuing institution.

Card and application issuer personnel function as administrators in the usual Common Criteria sense of the term.

B.2.1.5 General Model Use

The application is used in a transaction, which requires supporting software in a card acceptance device (CAD, often called a terminal). Most attacks are anticipated to occur at this stage of the life cycle. Much of the security required in the development environment is designed to protect against these attacks.

B.2.1.6 General Model End of Card Life

Typically, cards have an expiration date. The terminal will not accept a transaction from a card that has expired. SAMs are issued in relatively limited numbers and are usually under strict inventory control. They are returned to the issuer upon the end of card life. On the other hand, the end user card is seldom returned to the issuer but is usually simply discarded. Attackers may obtain discarded cards and use them to study the security features of similar cards still in use. If cryptographic keys provide some of the security, they must be securely managed. Generally, smart cards do not have the ability to destroy keys, other than session keys. Therefore expiration of keys must be carefully considered.

Applications have their own life cycles and can be variously instantiated on different cards and different types of cards. They do not typically have an end of life stage that is similar to the card's. They are either discontinued or evolve into the next version.

B.2.2 Soft Mask Card Life Cycle

A soft mask card, by definition, has an application added to the nonvolatile programmable memory. Table B.1 summarizes the soft mask card life cycle.

Table B.1 Soft Mask Card Life Cycle

Stage	Activities
User Requirements	• relatively small numbers of cards are needed, either because the customer base is small or because this is for a pilot
Design	• chip is designed to run many applications • operating system is designed to run many applications • application is designed with some knowledge of available operating systems and chips, but not necessarily all details • card design is dictated by user requirements

Stage	Activities
Manufacturing	• chip is fabricated with operating system in permanent memory • chip is tested • chip is packaged in a module that provides communication to off-chip world • packaged chip is tested • card is printed and inspected • packaged chip is embedded in card • application is added in nonvolatile programmable memory • card is tested again
Issuance	• completed card is delivered to issuer for finalization and personalization • card is tested • final step is to close off all test modes
Use	• card is used to conduct transactions • changes during use are possible
End of Life	• card reaches expiration date or is blocked • card is usually discarded • application design is re-evaluated in light of experience • application may be discontinued, continued on soft mask cards or taken to hard mask

B.2.2.1 Soft Mask Card User Requirements

Often the user does not know exactly what requirements might be appropriate and therefore decides to conduct a pilot in order to clarify them. This may have to do with what application(s) is most useful, how scalable the system must be, what commercial arrangements are acceptable, or any other aspect of the entire card system. A pilot is typically a limited deployment among relatively trusted end users (e.g., six months with a few hundred bank employees). Higher individual card costs are acceptable given the small number of cards involved.

Alternatively, the anticipated use of the card may involve relatively small numbers - on the order of tens of thousands per year. Such small runs do not warrant the high cost of having a dedicated mask made, particularly as the value of assets to be protected is likely to relatively low. User requirements may also be clear initially but may change over time. The added flexibility of a soft mask card is appropriate for these situations.

B.2.2.2 Soft Mask Card Design

Designing chips is expensive in terms of both time (1 to 2 years) and money. Consequently chip manufacturers typically design general purpose chips that will be useful to a variety of end users. Some of these may have no security requirements at all, while others may have relatively high security requirements. Consequently security features are often offered as options. The chip will be designed to support many different operating systems and many different applications. Marketing materials may describe the chip's capabilities and security options, which can provide information to potential attackers.

The operating system similarly is designed to support many different applications and perhaps to run on several different chips. It is not tailored to any specific application. Marketing materials will describe the operating system's capabilities. However, the operating system is generally programmed in the provider's proprietary language, which is known by a very few programmers, which is a security feature in itself.

The application(s) is designed with some knowledge of what general purpose chips and general purpose operating systems are available. It is typically designed to run on several chips and on different operating systems to avoid being dependent on any one single supplier. As not all of the user requirements are well known, a pilot is often used to clarify them. A soft mask card is chosen so that the application can be modified after issuance to address any problems that are found, without having to go back to the chip design stage.

In this case, the operating system and chip are not tailored to any particular application. The chip supplier may not even know what applications his chips are running. Design is less tightly coupled and security in the design environment may vary between the companies involved.

The application design may be made public so that knowledgeable persons can comment on and improve it.

B.2.2.3 Soft Mask Card Manufacturing

The chip is manufactured with the operating system in permanent memory. The chip is then tested, embedded in a module (package) and delivered to the card manufacturer. The card manufacturer independently prints the card and embeds the module in it. The application may be manufactured separately from the card and added to the nonvolatile programmable memory, usually by the card manufacturer.

There may be many different customers wanting different applications, all of which will run on the same chip and operating system. This permits the card manufacturer to customize his product to his customers without having to go back to the chip foundry for each one's needs. Economies of scale can be achieved between the card and chip manufacturer, while retaining the advantages of short production runs for the individual customers.

B.2.2.4 Soft Mask Card Issuance

The issuing institution generally takes delivery of the card and then personalizes it if the application requires that. The final step in personalization is usually to restrict the capacity for further changes, typically through the use of a cryptographic key.

B.2.2.5 Soft Mask Card Use

End users then use the card to conduct transactions. No changes to the card in use are anticipated, although there may be an ability to block an application or a card for further use. This is generally done only if it is reported lost or stolen. If any serious flaws in the design are discovered, or experience indicates that a change in the user requirements is advisable, the cards can be recalled and the application program can be patched.

B.2.2.6 Soft Mask Card End of Card Life

Typically the cards expire and are discarded. Any cryptographic keys used are usually changed so that expired cards can not be used to learn currently used keys. Expired cards can, however, be used to reverse engineer the application protocol, should an attacker so choose. If the application protocol was made public for comment earlier, application protocols would be generally known so reverse engineering would probably not be worth the expense and effort involved.

Many applications never go beyond this stage. Either the project is terminated or the total number of cards needed is not sufficient to warrant the expense of designing a hard mask. If the pilot was successful, the application will be redesigned if needed and than taken to hard mask.

B.2.2.7 Soft Mask Card Security Implications

The limited number of cards in use and the relatively short duration of a pilot limit the security exposure. Successful attacks may gain the attacker notoriety, but the assets being protected are limited and may make a successful attack more expensive than any financial benefit to be gained.

The ability to change the code is a security vulnerability, and special attention must be paid to the final step in personalization, which ensures subsequent changes can only be done under proper authorization.

Issuers must be aware that others are using the same chip and operating system in other, unknown applications. Some of these may fall in to the hands of attackers, who can reverse engineer them to gain knowledge to use in attacking other cards and other applications.

The general purpose chip and general purpose operating system are sold to a variety of users, and appropriate sales and marketing literature is prepared and distributed broadly. Detailed specifications for the chip and operating system, especially their security features, are usually available only under a nondisclosure agreement. However, the marketing materials may still be of use to potential attackers.

B.2.3 Hard Mask Card Life Cycle

If the project goes forward and a larger deployment is planned, the application may be taken to hard mask, which means the application is placed in permanent memory. This creates a new card life cycle, but is the second instantiation of the application.

Alternatively, the user requirements may be clear enough from the beginning to permit the development of a hard mask from the beginning of the project. Table B.2 summarizes the hard mask card life cycle.

Table B.2 Hard Mask Card Life Cycle

Stage	Activities
User Requirements	• relatively large numbers of cards are needed, and the application is well defined • no or very minor changes in use are anticipated
Design	• chip is customized to run this application • operating system is customized to run this application • application is designed with knowledge of operating systems and chips to be used, in all details • card design is dictated by user requirements
Manufacturing	• chip is fabricated with operating system and application in permanent memory • chip is tested • chip is packaged in a module that provides communication to off-chip world • packaged chip is tested • card is printed and inspected • packaged chip is embedded in card • card is tested again

Stage	Activities
Issuance	• completed card is delivered to issuer for finalization and personalization • card is tested • final step is to close off all remaining test modes
Use	• card is used to conduct transactions • very minor changes during use are possible.
End of Life	• card reaches expiration date or is blocked • card is usually discarded • application design may be re-evaluated in light of experience • application may be discontinued or continued on hard mask cards

B.2.3.1 Hard Mask Card User Requirements

The pilot may have clarified some user requirements or stimulated some new ones. An important part of a successful pilot is analyzing the results and deciding what changes need to be made.

B.2.3.2 Hard Mask Card Design

A redesign typically occurs since the hard mask card design normally is generated from a preceding pilot. This redesign may be because of changes in the user requirements or because a better, more efficient design has been worked out during the pilot period. The design may be optimized to run on a particular chip and operating system. Generally the chip chosen is a different one than the one used in the pilot, though usually of the same family. The soft mask card required enough nonvolatile programmable memory to store the entire application, while the hard mask card can place most of the application in permanent memory. Permanent memory is generally physically smaller and therefore cheaper (bit for bit) than nonvolatile programmable memory. The smallest possible chip is used, for reasons of economy, reliability, and security.

A larger deployment to a more diverse user base may require additional security. The application has been in use for a period of time (at least several months) and attackers may have obtained copies and begun the process of reverse engineering it.

B.2.3.3 Hard Mask Card Manufacture

The redesigned application is manufactured and delivered to the chip supplier for hard masking. Any chip options are selected. The chip is manufactured with the application in permanent memory. It can never be changed; what was software now becomes hardware. The chip is tested and then embedded in the module, then tested again. The module is then sent to the card manufacturer, who embeds it and then tests it again.

If the deployment is sufficiently large, there will be more than one manufacturer. Each manufacturer will try to protect its competitive advantage by incorporating its own intellectual property and additional features. These may be lower cost, faster speed, more security, or any other feature that will be attractive to the issuers.

B.2.3.4 Hard Mask Card Issuance

The application is personalized and issued, as was done with the soft mask card.

B.2.3.5 Hard Mask Card Use

In both soft and hard mask cards, the user is assumed not to be an information technology professional, and very limited flexibility and latitude for user input are provided. Consequently, there is usually no change in what the user sees or how he uses the card. If the application calls for it, the card can be blocked by the Issuer, possibly unblocked, and possibly have the PIN (assuming the application calls for such) changed.

If the application is used on a very broad scale, protection must be afforded against attacks based on knowledge derived from other instantiations of the application.

B.2.3.6 Hard Mask Card End of Life

When individual cards expire they are generally discarded, which means that there are increasing numbers of expired cards for potential attackers to study.

Applications typically do not end, however. They are either discontinued or evolve in to the next version. This means that any particular application that has been at issuance for some time must be aware that earlier, perhaps less secure, versions of it may still be around for attackers to learn from.

B.2.4 Proprietary Card Life Cycle

A user may decide to develop and deploy a proprietary chip and card that is unique to a particular system and not otherwise available. Table B.3 summarizes the proprietary card life cycle.

Table B.3 Proprietary Card Life Cycle

Stage	Activities
User Requirements	• relatively high security cards are needed • application is well defined
Design	• application, operating system, and chip are designed simultaneously and with full knowledge of each other • card design is dictated by user requirements
Manufacturing	• chip is fabricated with program code (which merges operating system and application) in permanent memory • chip is tested • chip is packaged in a module that provides communication to off-chip world • packaged chip is tested • card is printed and inspected • packaged chip is embedded in card • card is tested again
Issuance	• completed card is delivered to issuer for finalization and personalization (if application dictates) • card is tested • final step is to close off all remaining test modes
Use	• card is used to conduct transactions. • very minor changes during use are possible
End of Life	• card reaches expiration date or is blocked; unblocking may not be possible; card is usually returned to issuer. • application design may be re-evaluated in light of experience • application may be discontinued or continued on hard mask cards

B.2.4.1 Proprietary Card User Requirements

The user requirements will generally be based on earlier experience and will be carefully specified in a formal description of both technical and business requirements. These will include both functional and security requirements. These requirements drive the design.

B.2.4.2 Proprietary Card Design

The design of the program code (application plus operating system), chip and card will be carried on simultaneously, generally by a multi-company design team of specialists working in close coordination with each other. Cost is always a consideration, but security is likely to be high on the list of requirements. The application is typically coded and run on emulators and then on soft masked cards in a laboratory environment; the emulators and soft masked cards are never distributed to end users. Security in the development environment will be stringent.

B.2.4.3 Proprietary Card Manufacturing

The cards are hard masked (the application is placed in permanent memory) by the semiconductor manufacturer. These are usually single application cards, although multiple applications are possible. Manufacturing security is generally stringent. The chip is tested, embedded in the module, tested again, and delivered to the card manufacturer. The card manufacturer prints the plastic card, embeds the module in the card, and tests it again. There is generally no addition of other applications done by the card manufacturer. Delivery to the issuer is also under tight security.

B.2.4.4 Proprietary Card Issuance

The card may or may not be personalized by the issuer (there is no difference here between the card issuer and the application issuer). If personalized, it may be done by the issuer directly or by a personalization bureau that offers this service. This process is similar to enrolling a new employee in a company’s computer system. If it is a stored value card, the issuer may load value on it before delivering it to the end-user.

B.2.4.5 Proprietary Card Use

The end user uses the card in transactions in accordance with the terms of the agreement with the issuer. Most attacks are focused on this stage and most security arrangements are designed to address threats here.

B.2.4.6 Proprietary Card End of Life

Most such cards have an expiration date. Upon expiration, the card may be returned to the issuer or may be discarded. Attackers may obtain expired and/or discarded cards and use them to learn the security features of similar cards still in use. Disposition of the expired cards will be set by the initial user requirements.

B.2.4.7 Proprietary Card Security Implications

High security is typically a main user requirement, as these are always high value cards relative to other ones. These are theoretically the most secure smart cards, but that will depend on the extent to which security features were designed in and the design was properly executed.

B.2.5 Reconfigurable Card Life Cycle

By definition, a reconfigurable card has the capability of having applications added and deleted during usage. When the application can be loaded to reconfigurable cards, new possibilities for attack are created. Reconfigurable operating systems must anticipate attacks that may be based on other instantiations of the same application in less secure cards. Table B.4 summarizes the reconfigurable card life cycle.

Table B.4 Reconfigurable Card Life Cycle

Stage	Activities
User Requirements	• relatively large numbers of cards are needed • many different applications are anticipated • user requires ability to add and delete entire applications during usage
Design	• chip is designed to run the reconfigurable operating system • operating system is designed to support many applications, which are reconfigurable • application is designed with knowledge of operating systems and chips to be used, in all details • card design is dictated by user requirements
Manufacturing	• chip is fabricated with operating in permanent memory; may or may not include an application in permanent memory • chip is tested • chip is packaged in a module that provides communication to off-chip world • packaged chip is tested • card is printed and inspected • packaged chip is embedded in card • card is tested

Stage	Activities
Issuance	• completed card is delivered to issuer for finalization and personalization • additional application(s) may or may not be added in nonvolatile programmable memory • card is tested • final step is to close off all remaining test modes
Use	• card is used to conduct transactions • entire applications can be added or deleted • any applications in permanent memory can be logically blocked but not deleted
End of Life	• card reaches expiration date or is blocked; card is usually discarded • application design may be re-evaluated in light of experience • application may be discontinued, modified or reissued on other reconfigurable cards

B.2.5.1 Reconfigurable Card User Requirements

A principal user requirement is for the defining functionality: to be able to change the applications during the useful life of the card. This requires appropriate security on the program management functions, as the card will now be subject to virus-like attacks. A complete operating system will be needed, as the applications desired in the future are presently unknown.

Users want a very broad range of applications and of application providers. They particularly want the ability to take a new idea for an application and very quickly bring it to market. With the other types of cards, time to market is generally around a year, more if the plan includes a pilot stage. With reconfigurable cards and local application providers, a new application can theoretically be ready in one to several months.

B.2.5.2 Reconfigurable Card Design

The operating systems that will currently support these cards are Java^{TM}, Windows for Smart Cards^{TM}, or MEL^{TM} (the MULTOS^{TM} programming language). Each was designed by companies other than the conventional chip and card manufacturers, although there was much dialogue between them and the operating system suppliers. One of the advantages of these cards is that the programming languages are well known and many programmers can create applications for them. This means that the applications will be designed entirely separately from the chip, operating system, and card designs. It also means that there are many more programmers with the language skills necessary to attack these cards.

When these cards were first brought to market they used existing general purpose chips. As they mature, chips are being designed specifically to run each of them.

B.2.5.3 Reconfigurable Card Manufacturing

The chips, operating systems, cards and applications are manufactured separately. Operating systems are generally added to the chip at the chip manufacturing stage. Chips are embedded in modules and modules embedded in cards in the same way as with other cards, including the same testing. Manufacturing security is aimed primarily at maintaining companies' intellectual property, as the details of the designs and implementations must be widely known for effective programming to be done.

Applications are separately written and manufactured. They are expected to be loaded during the usage phase.

B.2.5.4 Reconfigurable Card Issuance

Reconfigurable cards are personalized to the end user and issued in the same way as other types of cards are. These cards raise the possibility of separate personalization of the card and the application(s) on it. The card can initially be issued with no application at all.

Instruction must be given to the end user on the proper loading and deletion of applications, which is not needed with the other types of cards. The end user has more options available than with the other types of cards, and hence more opportunity to make mistakes.

B.2.5.5 Reconfigurable Card Use

As with the other types of cards, these are used to conduct transactions. Applications must be properly loaded in order to work, and each application will have its own usage instructions and requirements. Security requirements for these must be stated in appropriate application protection profiles.

B.2.5.6 Reconfigurable Card End of Life

These cards are just coming to market and thus there is no experience with the end of the card life cycle.

B.2.6 Application Life Cycle

Any particular application typically goes through several card life cycles. In the simplest case, it is only instantiated on a proprietary card. In the more typical case, it is first instantiated on a soft mask card, then on a hard mask card, and ultimately on a reconfigurable card.

B.2.7 Multiple Application Cards

Some cards carry more than one application; this is becoming the usual practice. Such cards may have all applications in nonvolatile programmable memory, all in permanent memory, or some applications in permanent memory and some in nonvolatile programmable memory. They may thus be hard masked cards with respect to one or more applications, and soft masked cards with respect to others.

B.2.8 Complexities of the Real World

If an application is successful, it can mean that in some areas there are soft mask pilots in progress, in others that cards have gone to hard mask, and in still others that reconfigurable cards are being loaded with the application.

In addition, any particular card may have multiple applications, some of which are hard masked and others soft masked, and application separation is critically important to maintain the security of any one of them.

B.3 CC Impact

This section is intended to highlight those areas of smart card technology that impact directly upon the Common Criteria concepts. It is intended to assist both evaluators and authors of application protection profiles and security targets that use the SCSUG Smart Card Protection Profile through further explanation of the intent of the PP authors.

B.3.1 Terminology

Developer must distinguish between the chip developer, operating system developer, card developer, and application developer. Each must be responsible for the integrated product he provides to the next step in the development chain, or in the case of the application developer, for the proper functioning of the application if installed according to instructions.

Life Cycle may have multiple definitions. It may apply, as discussed in the sections above, to the sequence of fabrication and delivery of the smart card product. In this case, the elements of the life cycle will typically refer to physical manifestations of the product such as silicon or software. Life cycle may alternately be applied in the manner of ALC_LCD in which the elements will refer to the process of design, independent of the actual physical product. This life cycle definition would then deal with the sequence of specification, design review, code review, process management, review for production, etc. These two life cycle definitions thus have two distinctly different references.

Roles in Common Criteria primarily apply to developer, administrator, and user. In some smart card applications, these roles can become confused since there may be roles that are administrator in some senses and user in others. Examples are bank/issuer clerk, merchant clerk, doctor, nurse, pharmacist, etc. Each application must therefore clearly specify its roles and the privileges associated with those roles, and map these to the appropriate common criteria components.

User means variously the issuing institution, the end user, or the application. The issuing institution is the consumer who buys the product, while the end user is the person who uses the card and application to conduct a transaction. However, in the case of a multi application card, the user is more properly defined as an application resident on the card. In this case, the end users are not visible to the TOE. It sees and manages applications that themselves perform business functions.

User Data may mean end user (card holder) data.. Alternatively, if the TOE is being supplied for reconfigurable use, user data can equally apply to the programs or applications loaded onto the TOE as well as the final data important to the individual customer.

B.3.2 Operations

A variety of operations are possible for an SCSUG-SCPP compliant TOE. These are described in the main body of this protection profile. The following operations are a restatement of information already contained in the protection profile but are offered as additional interpretation of the detailed requirements.

- File and data access rights will be defined and only certain roles will be granted access privileges. This detailed definition shall be provided in the ST.
- Access conditions, once set, shall apply to all access and shall never be downgraded.
- The process and commands for creating the application file structure, including file access conditions, shall be controlled by access control provisions that are used only for this purpose. These provisions shall be detailed in the ST. The file structure for an application, once created, may be locked from any future modification or deletion.
- The platform must be capable of securely storing PIN and/or other secure data, including cryptographic data, using access control provisions that ensure that such data cannot be read from outside unless so authorized.
- No access to memory shall be allowed except as mediated through the card operating system.

- If blocking of the TOE is provided, it must prevent access to all functions by the cardholder and any entity other than that defined by the operating system. Card blocking should be reversible only by an authorized administrator.
- If disabling of the TOE is provided, such disablement shall prevent any further use and shall be non-reversible.
- Applications must be physically and/or logically separated from each other. The TSF must deny any information flow between applications except when specifically authorized.

B.3.3 Security Functional Requirements

Class FCS - Cryptographic support

Components of this class must address any security issues arising from the end of life cycle practices of a card. (This may involve expiring keys, having diversified keys, etc.).

FPT_SEP - Domain separation

Application separation may be achieved through either hardware or software or the synergistic operation of both. In a reconfigurable card, there may be a separate security domain for the card issuer and each application issuer, or an application issuer may have a security domain that provides cryptographic services to several applications. Security Targets must clearly document what security domains are present and what functionality each has.

B.3.4 Security Assurance Requirements

ADO_DEL - Delivery

This must be separately addressed for the delivery of the operating system to the chip, the embedded module to the card developer, and the application to the platform. Delivery of the application to the platform is significantly different for soft mask, hard mask, and post issuance addition and for deletion of applications to reconfigurable cards.

ADO_IGS - Installation, Generation and Start-up

This will be different depending on whether the entity doing the installation is the chip manufacturer installing an application on a hard mask card, the card manufacturer following the OS manufacturer's instructions to initialize the card, or the end user loading an application on to a reconfigurable card. All parts of the delivery require attention.

Class ADV - Development

Components of this class must address the development environment for the chip, operating system, and application separately if they are developed separately. They also must incorporate all functional specifications used for the chip, operating system, application(s) and card. These may include such high level specifications as EMV, GSM, etc.

AGD_ADM - Administrator Guidance

This guidance must include personalization instructions if a reconfigurable card or an application or both are to be personalized. Security features that are only enabled at the end of personalization must work properly if the administrator guidance is followed.

AGD_USR - User Guidance

User guidance must include a template of instructions to the end user that the issuing institution can combine with explanations of the agreement under which the card is issued. These may need to be translated in to various languages and may need to be tailored to the requirement of varying legal jurisdictions.

Class ALC - Life Cycle Support

The components of this class must detail which card life cycle is being used and, if an application, where in the application life cycle it is. ALC_DVS must reflect the user requirements for the card or application if it includes life cycle requirements.

AVA_VLA - Vulnerability Analysis

The vulnerability analysis must reflect the fact that the same TOE may be used in insecure applications as well as for more secure ones. A chip from an insecure use can be reverse engineered and the knowledge gained can be used to shorten the amount of vulnerability identification time. Marketing information on both the chip and the operating system can similarly be used to reduce the time needed to reverse engineer a card. High level application specifications such as EMV are public and can have a similar use. Soft mask cards may be in circulation for months before a hard mask card is designed. The question becomes one of when the vulnerability is initially attacked. As the Common Evaluation Methodology makes time an important element in calculating vulnerability assessment (and attack potential), a critical question is when “the clock starts ticking”. If a hard mask card has been preceded by a soft mask card pilot and the chip chosen is also used in a variety of other applications, it must be assumed that attackers have studied both and have reverse engineered most of the chip and the application.

B.3.5 Evaluation

Physical Attacks

These considerations dictate that evaluation procedures depend not only on software engineering, cryptography, and hardware engineering, but also include the relationships between them. Evaluation facilities should be competent in all of these areas, and particularly in areas concerning the interrelationship issues.

(This page purposely left blank)

Annex C - Security Functional Requirements Support

C.1 Management of Functions in TSF

Components FMT_MOF.1 (Management of security functions behavior), FMT_MSA.1 (Management of security attributes), and FMT_MTD.1 (Management of TSF data) allow certain authorized roles to manage the behavior of TSF functions that use rules or have specified conditions that may be manageable. Table C.1 lists the IT security functional components which have a management function applied to the SCSUG-SCPP compliant TOE. These management functions were derived from the unique characteristics of the TOE and from a review of the actions which could be considered for the management functions listed for consideration in the Common Criteria Part 2.

Following Table C.1, each requirement is listed, along with the related information from the CC, Part 2, regarding management actions for consideration. An explanation is provided regarding why each requirement was or was not chosen for specific inclusion in the management functions.

Table C.1 Security Functional Components Management Options

Component	Component Name	Management Functions
FAU_ARP.1	Security alarms	FMT_MOF.1 (b)
FAU_SAA.1	Potential violation analysis	FMT_MOF.1 (c)
FCS_CKM.1	Cryptographic key generation	FMT_MOF.1 (d)
FCS_CKM.3	Cryptographic key access	FMT_MOF.1 (d)
FCS_CKM.4	Cryptographic key destruction	FMT_MOF.1 (d)
FCS_COP.1	Cryptographic operation	FMT_MOF.1 (d)
FDP_ACF.1	Security attribute based access control	FMT_MOF.1 (a)
FDP_IFF.1	Simple security attributes	FMT_MOF.1 (a)
FIA_AFL.1	Authentication failure handling	FMT_MOF.1 (e) FMT_MTD.1 (a)
FIA_UAU.1	Timing of authentication	FMT_MOF.1 (f)
FIA_UID.1	Timing of identification	FMT_MOF.1 (g)

Component	Component Name	Management Functions
FMT_REV.1	Revocation	FMT_MOF.1 (h)
FPT_RPL.1	Replay detection	FMT_MOF.1 (i)
FPT_TST.1	TSF testing	FMT_MOF.1 (j)
FRU_RSA.1	Maximum quotas	FMT_MOF.1 (k)

C.1.1 Management Actions for Consideration

Management actions could be considered from the following functions. The first statement given below each security functional requirement is that provided in ISO 15408, Part 2, as suggested management activities. The second statement relates these suggestions to the TOE, identifying how they have been incorporated into the SCSUG-SCPP.

FAU_ARP.1 - Security alarms

The following action could be considered for the management functions in FMT Management:

a) the management of ***actions to be taken*** in the event of a security alarm.

The TOE may be provided with the ability to respond in a variety of ways to an identified security alarm. This is therefore a potential management function and is accordingly included as item (b) in FMT_MOF.1.

FAU_SAA.1 - Potential violation analysis

The following action could be considered for the management functions in FMT Management:

a) maintenance of the violation analysis rules by ***adding, modifying, or deleting rules*** from the set of rules.

The TOE may be provided with the ability to identify differing potential violations. This is therefore a potential management function and is accordingly included as item (c) in FMT_MOF.1.

FCS_CKM.1 - Cryptographic key generation

FCS_CKM.3 - Cryptographic key access

FCS_CKM.4 - Cryptographic key destruction

The following action could be considered for the management functions in FMT Management:

a) the management of ***changes to cryptographic key attributes***. Examples of key attributes include user, key type (e.g. public, private, secret), validity period, and use (e.g. digital signature, key encryption, key agreement, data encryption).

The TOE may be provided with alternate forms of encryption which could be selected post-issuance. This is therefore a potential management function and is accordingly included as item (d) in FMT_MOF.1.

FCS_COP.1 - Cryptographic operation

There are no management activities foreseen for this component.

In contrast to the CC suggestion that there are no management activities, the TOE may be provided with alternate forms of encryption which could be selected post-issuance. This is therefore a potential management function and is accordingly included as item (d) in FMT_MOF.1.

FDP_ACF.1 - Security attribute based access control

The following action could be considered for the management functions in FMT Management:

a) managing the attributes used to make explicit access or denial based decisions.

The TOE may be provided with the ability to allow for post-issuance modification of the access control functions. This is therefore a potential management function and is accordingly included as item (a) in FMT_MOF.1. It is also important to note that no management actions can be allowed to reduce the attributes required for access.

FDP_IFF.1 - Simple security attributes

The following action could be considered for the management functions in FMT Management:

a) managing the attributes used to make explicit access based decisions.

The TOE may be provided with the ability to allow for post-issuance modification of the information flow control functions. This is therefore a potential management function and is accordingly included as item (a) in FMT_MOF.1. It is also important to note that no management actions can be allowed to reduce the attributes required for access.

FIA_AFL.1 - Authentication failure handling

The following actions could be considered for the management functions in FMT Management:

a) management of the ***threshold for unsuccessful authentication attempts***;
b) management of ***actions to be taken in the event of an authentication failure***.

Item (a) could be managed under FMT_MTD (Management of TSF data) and is accordingly included as item (a) in FMT_MTD.1. Item (b) could be managed under FMT_MOF (Management of security functions behavior) and is accordingly included as item (e) in FMT_MOF.1.

FIA_UAU.1 - Timing of authentication

The following actions could be considered for the management functions in FMT Management:

a) management of the authentication data by an administrator;

b) management of the authentication data by the associated user;

c) managing the list of ***actions that can be taken before the user is authenticated.***

Item (c) could be managed under FMT_MOF (Management of security functions behavior). It is accordingly included in FMT_MOF.1 as item (f).

FIA_UID.1 - Timing of identification

The following actions could be considered for the management functions in FMT Management:

a) the management of the user identities;

b) if an authorized administrator can ***change the actions allowed before identification, the managing of the action lists***.

Item (b) could be managed under FMT_MOF (Management of security functions behavior). It is accordingly included in FMT_MOF.1 as item (g).

FMT_REV.1 - Revocation

The following actions could be considered for the management functions in FMT Management:

a) managing the group of roles that can invoke revocation of security attributes;

b) managing the lists of users, subjects, objects and other resources for which revocation is possible;

c) managing the ***revocation rules***.

Item (c) could be managed under FMT_MOF (Management of security functions behavior). It is accordingly included in FMT_MOF.1 as item (h).

FPT_RPL.1 - Replay detection

The following actions could be considered for the management functions in FMT Management:

a) management of the list of identified entities for which replay shall be detected;

b) management of the ***list of actions that need to be taken in case of replay***.

Item (b) could be managed under FMT_MOF (Management of security functions behavior). It is accordingly included in FMT_MOF.1 as item (i).

FPT_TST.1 - TSF testing

The following actions could be considered for the management functions in FMT Management:

a) management of the ***conditions under which TSF self testing occurs***, such as during initial start-up, regular interval, or under specified conditions;

b) management of the time interval if appropriate.

Item (a) could be managed under FMT_MOF (Management of security functions behavior). It is accordingly included in FMT_MOF.1 as item (j).

FRU_RSA.1 - Maximum quotas

The following action could be considered for the management functions in FMT Management:

a) management of ***maximum limits for a resource for groups and/or individual users and/or subjects by an administrator***.

The TOE may be provided with the ability to add and delete additional functions and applications which could use resources. This is therefore a potential management function and is accordingly included as item (k) in FMT_MOF.1.

C.1.2 No Management Actions Foreseen

There are no management actions foreseen for the following functions. The first statement given below each security functional requirement is that provided in ISO 15408, Part 2, as suggested management activities. The second statement relates these suggestions to the TOE, identifying how they have been incorporated into the SCSUG-SCPP.

FAU_LST.1 - Audit list generation

There are no management activities foreseen.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of an integral function such as selection of audit list candidates. No management actions are therefore identified.

FAU_SEL.1 - Selective audit

The following action could be considered for the management functions in FMT Management:

a) maintenance of the rights to view/modify the audit events.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of an integral function such as audit lists of specific events. No management actions are therefore identified.

FAU_STG.1 - Protected audit trail storage

There are no management activities foreseen.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of an integral function such as audit lists of specific events. No management actions are therefore identified.

FAU_STG.4 - Prevention of audit data loss

The following action could be considered for the management functions in FMT Management:

a) maintenance (deletion, modification, addition) of actions to be taken in case of audit storage failure.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of an integral function such as selection of audit list operations. No management actions are therefore identified.

FDP_ACC.1 - Subset access control

There are no management activities foreseen for this component.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the access control functions. No management actions are therefore identified.

FDP_ETC.1 - Export of user data without security attributes

There are no management activities foreseen for this component.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the requirements for export of user data. No management actions are therefore identified.

FDP_IFC.1 - Subset information flow control

There are no management activities foreseen for this component.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the information flow control functions. No management actions are therefore identified.

FDP_ITC.1 - Import of user data without security attributes

The following action could be considered for the management functions in FMT Management:

a) the modification of the additional control rules used for import.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the requirements for import of user data. No management actions are therefore identified.

FDP_ITT.1 - Basic internal transfer protection

The following action could be considered for the management functions in FMT Management:

a) if the TSF provides multiple methods to protect user data during transmission between physically separated parts of the TOE, the TSF could provide a pre-defined role with the ability to select the method that will be used.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the mechanisms used for internal data transfer protection. No management actions are therefore identified.

FDP_RIP.1 - Subset residual information protection

The following action could be considered for the management functions in FMT Management:

a) the choice of when to perform residual information protection (i.e. upon allocation or de-allocation) could be made configurable within the TOE.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the mechanisms for protection of residual data. No management actions are therefore identified.

FDP_UIT.1 - Data exchange integrity

There are no management activities foreseen for this component.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the mechanisms for maintenance of data exchange integrity. No management actions are therefore identified.

FIA_ATD.1 - User attribute definition

The following action could be considered for the management functions in FMT Management:

a) if so indicated in the assignment, the authorized administrator might be able to define additional security attributes for users.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the attribute definition tables. No management actions are therefore identified.

FIA_UAU.7 - Protected authentication feedback

There are no management activities foreseen.

It would be ill-advised in the operating environment of the TOE to ever provide information to the user during the process of authentication. No management actions are therefore identified.

FMT_MOF.1 - Management of security functions behavior

The following action could be considered for the management functions in FMT Management:

a) managing the group of roles that can interact with the functions in the TSF.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the capabilities assigned to each role. No management actions are therefore identified.

FMT_MSA.1 - Management of security attributes

The following action could be considered for the management functions in FMT Management:

a) managing the group of roles that can interact with the security attributes.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the capabilities assigned to each role. No management actions are therefore identified.

FMT_MSA.2 - Secure security attributes

There are no additional management activities foreseen for this component.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the definitions and limits on secure values. No management actions are therefore identified.

FMT_MSA.3 - Static attribute initialization

The following actions could be considered for the management functions in FMT Management:

a) managing the group of roles that can specify initial values;

b) managing the permissive or restrictive setting of default values for a given access control SFP.

The TOE, once issued, will have all initial values set. Any additional initialization will be included in post-issuance installation of new capabilities. These will also include any specific new initial values. Thus, there is no useful function served by allowing modification of the roles or settings for initialization post-issue.

FMT_MTD.1 - Management of TSF data

The following action could be considered for the management functions in FMT Management:

a) managing the group of roles that can interact with the TSF data.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the capabilities assigned to each role. No management actions are therefore identified.

FMT_MTD.2 - Management of limits on TSF data

The following action could be considered for the management functions in FMT Management:

a) managing the group of roles that can interact with the limits on the TSF data.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the capabilities assigned to each role. No management actions are therefore identified.

FMT_MTD.3 - Secure TSF data

There are no additional management activities foreseen for this component.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the definitions of secure values. No management actions are therefore identified.

FPT_FLS.1 - Failure with preservation of secure state

There are no management activities foreseen.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the operations in the event of failures. No management actions are therefore identified.

FPT_ITI.1 - Inter-TSF detection of modification

There are no management activities foreseen.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance changes in the capability to detect and respond to modification in TSF data exchanged with another trusted IT product. No management actions are therefore identified.

FPT_ITT.1 - Basic internal TSF data transfer protection

The following actions could be considered for the management functions in FMT Management:

a) management of the types of modification against which the TSF should protect;

b) management of the mechanism used to provide the protection of the data in transit between different parts of the TSF.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance changes in the capability to protect data from modification when it is passed between separate parts of the TOE. No management actions are therefore identified.

FPT_PHP.3 - Resistance to physical attack

The following action could be considered for the management functions in FMT Management:

a) management of the automatic responses to physical tampering.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the actions to be taken in the event of physical tampering. No management actions are therefore identified.

FPT_RCV.3 - Automated recovery without undue loss

The following actions could be considered for the management functions in FMT Management:

a) management of who can access the restore capability within the maintenance mode;

b) management of the list of failures/service discontinuities that will be handled through the automatic procedures.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the automated recovery operations. No management actions are therefore identified.

FPT_RCV.4 - Function recovery

There are no management activities foreseen.

Smart card capabilities are not, in general, extensive enough to allow for post-issuance modification of the function recovery operation. No management actions are therefore identified.

FPT_RVM.1 - Non-bypassability of the TSP

There are no management activities foreseen.

Smart card capabilities do not introduce any new considerations on non-bypassability which would be manageable. No management actions are therefore identified.

FPT_SEP.1 - TSF domain separation

There are no management activities foreseen.

Smart card capabilities do not introduce any new considerations on domain separation which would be manageable. No management actions are therefore identified.

C.2 Functional Component Operations

The components selected for application in the SCSUG-SCPP have a set of operations which must be fully specified in a compliant ST. Some of these operations have been completed in this PP. These completions represent the general applicability of the TOE suitable for financial services applications. Other operations require a specificity dependent on the details of intended application. These are left for completion in the ST. The following sections provide the details of each of these sets of operations. This information is presented in Section 5 (IT Security Requirements) but is reformatted here to provide a clear reference for the writer of a compliant ST.

C.2.1 Operations Completed in the PP

A number of operations have been specified in this PP. The following list compiles these operations.

The element **FAU_LST.1.1 (Audit list generation)** has a partially completed operation regarding the specification of auditable events. Additional events may also be specified.

[assignment: specifically defined auditable events] is completed as:

a) production history file

The element **FAU_LST.1.2 (Audit list generation)** has a partially completed operation regarding the information to be preserved. Additional production history events may also be specified.

[assignment: specifically defined auditable events] is completed as:

a) production history file shall contain:

1. IC type and fabricator

2. IC fabrication date and batch identifier

3. IC serial number

4. Operating software identification and release date

5. IC Module fabricator and packaging date

6. ICC manufacturer and embedding date

7. IC prepersonalization equipment and date

8. *other specifically defined history events*

The element **FAU_STG.1.2 (Protected audit trail storage)** has a completed operation regarding modifications to the audit records:

[selection: prevent, detect] is completed as **prevent**

The element **FAU_STG.4.1 (Prevention of audit data loss)** has a completed operation regarding the handling of the audit data list:

[selection: ignore auditable events, prevent auditable events, except those taken by the authorized user with special rights, overwrite the oldest stored audit records] is completed as: **overwrite the oldest stored audit records**

The element **FDP_ITT.1.1 (Basic internal transfer protection)** has a completed operation regarding the actions to be taken in the protection of user data:

[selection: disclosure, modification, loss of use] is completed as **disclosure or modification**

The element **FDP_RIP.1.1 (Subset residual information protection)** has a completed operation regarding the actions after which resource information is to be unavailable:

[selection: allocation of the resource to, deallocation of the resource from] is completed as **deallocation of the resource from**

The element **FDP_UIT.1.1 (Data exchange integrity)** has completed operations regarding the source of data being acted upon, and the protection offered:

[selection: transmit, receive] is completed as **transmit and receive**

[selection: modification, deletion, insertion, replay] is completed as **modification**

The element **FDP_UIT.1.2 (Data exchange integrity)** has a completed operation regarding effect to be looked for:

[selection: modification, deletion, insertion, replay] is completed as **modification**

The element **FIA_UAU.7.1 (Protected authentication feedback)** has a completed operation regarding the feedback allowed during authentication:

[assignment: list of feedback] is completed as **none**

The element **FMT_MOF.1.1 (Management of security functions behavior)** has a completed operation regarding the range of management actions allowed:

[selection: determine the behavior of, disable, enable, modify the behavior of] is completed as **modify the behavior of**

The element **FMT_MOF.1.1 (Management of security functions behavior)** has a partially completed operation regarding the list of functions to be managed. The list of functions is supplied. The details required for total specification remain to be completed.

[assignment: list of functions] is completed as:

a) **management of *data access levels*, which, once established, shall never be reduced**

b) **management of *actions to be taken* in the event of a security alarm**

c) **maintenance of the violation analysis rules by *adding, modifying, or deleting rules* from the set of rules**

d) **management of *changes to cryptographic key attributes* including key type (e.g. public, private, secret), validity period, and use (e.g. session key, digital signature, key encryption, key agreement, data encryption)**

e) **management of *actions to be taken in the event of an authentication failure***

f) **managing the *list of actions that can be taken before the user is authenticated***

g) **if an authorized administrator can change the *actions allowed before identification, the managing of the action lists***

h) **managing the *revocation rules***

i) **management of the *list of actions that need to be taken in case of replay***

j) **management of the *conditions under which TSF self testing occurs*, such as during initial start-up, at regular intervals, or under specified conditions**

k) **management of maximum quotas of resources which may be used**

l) management of additional ***list of functions*** to be detailed in the ST

The element **FMT_MSA.3.1 (Static attribute initialization)** has a completed operation regarding the nature of default values:

[selection: restrictive, permissive, other property] is completed as **restrictive**

The element **FMT_MTD.2.1 (Management of limits on TSF data)** has a partially completed operation regarding the list of TSF data to be managed. The list of TSF data is supplied. The details required for total specification remain to be completed.

[assignment: list of TSF data] is completed as:

a) **management of the *threshold for unsuccessful authentication attempts***

b) management of additional ***list of functions*** to be detailed in the ST

The element **FPT_ITT.1.1 (Basic internal TSF data transfer protection)** has a completed operation regarding the protection to be offered to TSF data:

[selection: disclosure, modification] is completed as **modification**

The element **FPT_PHP.3.1 (Resistance to physical attack)** has a completed operation regarding the applicable scenarios:

[assignment: physical tampering scenarios] is completed as **environmental stress**

The element **FPT_RCV.3.2 (Automated recovery without undue loss)** has a completed operation regarding the failures to be addressed:

[assignment: list of failures/service discontinuities] is completed as **power failure during operation**

The element **FPT_RCV.4.1 (Function recovery)** has a completed operation regarding the security functions and scenarios to be addressed:

[assignment: list of SFs and failure scenarios] is completed as **the security functions involved in rollback and reset functions and the scenario of power loss or smart card withdrawal prior to completion**

C.2.2 Operations Deferred to the ST

A number of operations can not be specified in this PP due to the anticipated breadth of application. The following operations are identified as requiring completion in the ST complying with this PP.

The element **FAU_ARP.1 (Security alarms)** has an incomplete operation regarding actions to be taken:

[assignment: list of the least disruptive actions]

The element **FAU_LST.1.1 (Audit list generation)** has a partially incomplete operation regarding the specification of auditable events. Certain history related events have been specified.

[assignment: specifically defined auditable events]

The element **FAU_LST.1.2 (Audit list generation)** has a partially incomplete operation regarding information to be preserved. Certain history related events have been specified.

[assignment: audit relevant information]

The element **FAU_SAA.1.2 (Potential violation analysis)** has incomplete operations regarding event selection:

[assignment: subset of defined auditable events]
[assignment: any other rules]

The element **FAU_SEL.1.1 (Selective Audit)** has incomplete operations regarding the attributes of the auditable events:

[selection: object identity, user identity, subject identity, host identity, event type]
[assignment: list of additional attributes that audit selectivity is based upon]

The element **FAU_STG.4.1 (Prevention of audit data loss)** has an incomplete operation regarding the handling of the audit data list.

[assignment: other actions to be taken in case of audit storage failure]

The element **FCS_CKM.1.1 (Cryptographic key generation)** has incomplete operations regarding mechanisms and specifications:

[assignment: cryptographic key generation algorithm]

[assignment: cryptographic key sizes]

[assignment: list of standards]

The element **FCS_CKM.3.1 (Cryptographic key access)** has incomplete operations regarding mechanisms and specifications:

[assignment: type of cryptographic key access]

[assignment: cryptographic key access method]

[assignment: list of standards]

The element **FCS_COP.1.1 (Cryptographic operation)** has incomplete operations regarding mechanisms and specifications:

[assignment: list of cryptographic operations]

[assignment: cryptographic algorithm]

[assignment: cryptographic key sizes]

[assignment: list of standards]

The element **FDP_ACC.1.1 (Subset access control)** has incomplete operations regarding the policies to be followed and the details of application:

[assignment: access control SFP]

[assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

The element **FDP_ACF.1.1 (Security attribute based access control)** has incomplete operations regarding the policies to be followed and the details of application:

[assignment: access control SFP]

[assignment: security attributes, named groups of security attributes]

The element **FDP_ACF.1.2 (Security attribute based access control)** has an incomplete operation regarding the details of application:

[assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

The element **FDP_ACF.1.3 (Security attribute based access control)** has an incomplete operation regarding additional access control rules:

[assignment: rules, based on security attributes, that explicitly authorize access of subjects to objects]

The element **FDP_ACF.1.4 (Security attribute based access control)** has an incomplete operation regarding additional access control rules:

[assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

The element **FDP_ETC.1.1 (Export of user data without security attributes)** has an incomplete operation regarding the assignment of an SFP:

[assignment: access control SFP(s) and/or information flow control SFP(s)]

The element **FDP_IFC.1.1 (Subset information flow control)** has incomplete operations regarding the policies to be followed and the details of application:

[assignment: information flow control SFP]

[assignment: list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP]

The element **FDP_IFF.1.1 (Simple security attributes)** has incomplete operations regarding the application policies to be followed and the applications of those policies:

[assignment: information flow control SFP]

[assignment: the minimum number and type of security attributes]

The element **FDP_IFF.1.2 (Simple security attributes)** has an incomplete operation regarding the details of application:

[assignment: for each operation, the security attribute-based relationship that must hold between subject and information security attributes]

The element **FDP_IFF.1.3 (Simple security attributes)** has an incomplete operation regarding additional rules:

[assignment: additional information flow control SFP rules]

The element **FDP_IFF.1.4 (Simple security attributes)** has an incomplete operation regarding additional capabilities:

[assignment: list of additional SFP capabilities]

The element **FDP_IFF.1.5 (Simple security attributes)** has an incomplete operation regarding additional rules:

[assignment: rules, based on security attributes, that explicitly authorize information flows]

The element **FDP_IFF.1.6 (Simple security attributes)** has an incomplete operation regarding the rules for denial of information flow:

[assignment: rules, based on security attributes, that explicitly deny information flows]

The element **FDP_ITC.1.1 (Import of user data without security attributes)** has an incomplete operation regarding the identification of SFPs:

[assignment: access control SFP and/or information flow control SFP]

The element **FDP_ITC.1.3 (Import of user data without security attributes)** has an incomplete operation regarding additional rules:

[assignment: additional importation control rules]

The element **FDP_ITT.1.1 (Basic internal transfer protection)** has an incomplete operation regarding the identification of SFPs:

[assignment: access control SFP(s) and/or information flow control SFP(s)]

The element **FDP_RIP.1.1 (Subset residual information protection)** has an incomplete operation regarding the objects for which the deallocation applies:

[assignment: list of objects]

The element **FDP_UIT.1.1 (Data exchange integrity)** has an incomplete operation regarding the identification of an SFP:

[assignment: access control SFP(s) and/or information flow control SFP(s)]

The element **FIA_AFL.1.1 (Authentication failure handling)** has incomplete operations regarding the number and type of events to be considered:

[assignment: number]

[assignment: list of authentication events]

The element **FIA_AFL.1.2 (Authentication failure handling)** has an incomplete operation regarding the action to be taken when required:

[assignment: list of actions]

The element **FIA_ATD.1.1 (User attribute definition)** has an incomplete operation regarding the specification of the individual user attributes to be maintained:

[assignment: list of security attributes]

The element **FIA_UAU.1.1 (Timing of authentication)** has an incomplete operation regarding which operations are allowed prior to authentication:

[assignment: list of TSF-mediated actions]

The element **FIA_UID.1.1 (Timing of identification)** has an incomplete operation regarding which operations are allowed prior to identification:

[assignment: list of TSF-mediated actions]

The element **FMT_MOF.1.1 (Management of security functions behavior)** has incomplete operations regarding the details of the functions to be included and the roles allowed to perform such modifications. The list of functions is completed. The details required for total specification remain to be completed:

[assignment: list of functions including

a) management of *data access levels*, which, once established, shall never be reduced
b) management of *actions to be taken* in the event of a security alarm
c) maintenance of the violation analysis rules by *adding, modifying, or deleting rules* from the set of rules
d) management of *changes to cryptographic key attributes* including key type (e.g. public, private, secret), validity period, and use (e.g. session key, digital signature, key encryption, key agreement, data encryption)
e) management of *actions to be taken in the event of an authentication failure*
f) managing the *list of actions that can be taken before the user is authenticated*
g) if an authorized administrator can change the *actions allowed before identification, the managing of the action lists*

h) managing the *revocation rules*

i) management of the *list of actions that need to be taken in case of replay*

j) management of the *conditions under which TSF self testing occurs*, such as during initial start-up, at regular intervals, or under specified conditions

k) management of maximum quotas of resources which may be used

l) management of additional *list of functions* to be detailed in the ST]

[assignment: the authorized identified roles]

The element **FMT_MSA.1.1 (Management of security attributes)** has incomplete operations regarding identification of SFPs, selection of specific attributes and restrictions on the ability to affect them:

[assignment: access control SFP, information flow control SFP]

[selection: change_default, query, modify, delete,]

[assignment: other operations]

[assignment: list of security attributes]

[assignment: the authorized identified roles]

The element **FMT_MSA.3.1 (Static attribute initialization)** has an incomplete operation regarding the identification of SFPs:

[assignment: access control SFP, information flow control SFP]

The element **FMT_MSA.3.2 (Static attribute initialization)** has an incomplete operation regarding the roles allowed to specify default values:

[assignment: the authorized identified roles]

The element **FMT_MTD.1.1 (Management of TSF data)** has incomplete operations regarding selection of specific data and restrictions on the ability to affect it:

[selection: change_default, query, modify, delete, clear,]

[assignment: other operations]

[assignment: list of TSF data]

[assignment: the authorized identified roles]

The element **FMT_MTD.2.1 (Management of limits on TSF data)** has incomplete operations regarding selection of specific data and restrictions on the ability to affect it:

[assignment: list of TSF data including:

a) management of the *threshold for unsuccessful authentication attempts*;

b) management of additional *list of functions* to be detailed in the ST]

[assignment: the authorized identified roles]

The element **FMT_MTD.2.2 (Management of limits on TSF data)** has an incomplete operation regarding the action when limits are at or exceeded:

[assignment: actions to be taken]

The element **FMT_REV.1.1 (Revocation)** has incomplete operations regarding who may revoke specified security attributes:

[selection: users, subjects, objects, other additional resources]
[assignment: the authorized identified roles]

The element **FMT_REV.1.2 (Revocation)** has an incomplete operation regarding the rules for revocation:

[assignment: specification of revocation rules]

The element **FPT_FLS.1.1 (Failure with preservation of secure state)** has an incomplete operation regarding the failures for which preservation of secure state is required:

[assignment: list of types of failures in the TSF]

The element **FPT_ITI.1.1 (Inter-TSF detection of modification)** has an incomplete operation regarding the metric used to detect such modification:

[assignment: a defined modification metric]

The element **FPT_ITI.1.2 (Inter-TSF detection of modification)** has an incomplete operation regarding the action to be taken when modification is detected:

[assignment: action to be taken]

The element **FPT_PHP.3.1 (Resistance to physical attack)** has an incomplete operation regarding the list of TSF devices which are required to withstand the specified (environmental stress) attack:

[assignment: list of TSF devices/elements]

The element **FPT_RCV.3.3 (Automated recovery without undue loss)** has an incomplete operation regarding specification of the allowed loss upon recovery:

[assignment: quantification]

The element **FPT_RPL.1.1 (Replay detection)** has an incomplete operation regarding which entities shall be monitored for replay:

[assignment: list of identified entities]

The element **FPT_RPL.1.2 (Replay detection)** has an incomplete operation regarding the actions to be taken upon detection of replay:

[assignment: list of specific actions]

The element **FPT_TST.1.1 (TSF Testing)** has an incomplete operation regarding conditions when the self test should be run:

[assignment: conditions under which self test should occur]

The element **FRU_RSA.1 (Maximum Quotas)** has incomplete operations regarding the subjects, users, and limitations for imposition of quota limits:

[assignment: resources to be controlled]
[selection: individual user, defined group of users, subjects]
[selection: simultaneously, over a specified period of time]

The element **FTP_ITC.1.2 (Inter-TSF trusted channel)** has an incomplete operation regarding the initiation of communications:

[selection: the TSF, the remote trusted IT product]

The element **FTP_ITC.1.3 (Inter-TSF trusted channel)** has an incomplete operation regarding the specification of functions for which the secure channel is necessary:

[assignment: list of functions for which a trusted channel is required]

Annex D Packages

D.1 Introduction

The SCSUG-SCPP intends to identify and set forth a comprehensive and reusable collection of smart card security requirements. As such, this protection profile applies to an integrated product produced by potentially different manufacturers and developers. The Common Criteria states, however, that:

> The case where an ST claims to be partially conformant to a PP is not admissible for CC evaluation (ISO 15408-1, page 49).

Therefore, although this PP is to be applied as a single PP against a single integrated product, developers may find considering the use of packages based on this PP to be helpful. In this manner, components of the overall smart card system could individually be evaluated in a manner which would be directly supportive of the final evaluation of the integrated system. This annex addresses the development of one potential set of packages which could be amplified as required for this use.

D.2 CC Definition of Packages

The Common Criteria defines packages in this manner:

> An intermediate combination of components is termed a package. The package permits the expression of a set of functional or assurance requirements that meet an identifiable subset of security objectives. A package is intended to be reusable and to define requirements that are known to be useful and effective in meeting the identified objectives. A package may be used in the construction of larger packages, PPs, and STs. (ISO 15408-1, page 26).

The intent is thus that these SCSUG-SCPP–defined packages be utilized as reusable support in the development of appropriate component security targets. If directly applied in this manner, these packages could significantly contribute to simplifying the evaluation of the final integrated product based on this PP.

D.3 SCPP Package Concepts

The packages defined here are a combination of requirements (SFRs and SARs) that reflect likely real world products that could be considered as TOEs. Each defined package is associated with one or more threats or policies with associated objectives.

The basic physical component of the TOE is an integrated circuit chip, as a chip vendor would market it to a card vendor. The chip has a physical structure that may provide some protection against certain types of attacks. It constitutes the physical manifestation of the final product and, as such, requires evaluation of physical structure.

An SCSUG-compliant chip will contain some form of operating software that provides functionality allowing a secure communication channel to a trusted source. Other functionality may be supplied as well. The software, which may be mask programmed, added in EEPROM to the basic chip, or supplied as a post-issuance download, may be developed separately from the chip and can be functionally tested for operational functions and some error performance without installation onto the chip. For the purposes of this package definition, it is assumed that the software is considered separately from the integrated circuit. Thus, certain functions which may require the details of the IC implementation will not be fully evaluated until the integrated platform is evaluated.

The integrated platform represents the combination of the ICC and the operating software into a single unified package and is the subject of this protection profile. This final combination of components particularly requires the evaluation of those parts of performance which are synergistic between hardware and software or of those parts which depend on the operational integrity of the final unit. Thus, the final security evaluation must be done with the TOE software on the operational chip, as the synergy between the chip and the software supports the required security.

Thus, the packages which could be directly, logically derived from this PP include:

- integrated circuit (IC)
- operating software (OS)
- integrated platform (SCSUG-SCPP)

D.4 Package Use

Figures D.1, D.2, D.3, and D.4 illustrate the anticipated use and utility of packages.

In Figure D.1, the SCSUG-SCPP is referenced as a single entity in the development of a Security Target for a conformant TOE. The ST then claims compliance with the SCSUG-SCPP. The TOE and the supporting evaluation evidence are presented to the evaluation laboratory for review and determination of compliance. This evaluation depends on components and input from all developers involved in the integrated TOE, and can not be performed until the final instantiation of the product is complete. This is the simplest application of the SCSUG-SCPP, but requires inputs from all of the developers involved in generating the total system.

Figures D.2 and D.3 illustrate the alternative use of packages to allow intermediate evaluations with reuse of results for IC and OS respectively. In this use, it is anticipated that the packages defined below would be included in the statement of a component ST. This could be done either through direct restatement of the packages as one of the inputs or through demonstration of conformance to the requirements stated in the package with an alternate set of inputs (e.g., from another protection profile). Along with supporting evidence, the component TOE could then be furnished to an evaluation lab for certification review. Any certificate issued would not be based on the SCSUG-SCPP since partial conformance to a PP is not allowed. However, the component might claim compliance to any other fully incorporated PP. Reference would be made in the ST, in any case, to the inclusion of the package as defined in the SCSUG-SCPP.

Figure D.4 then illustrates the final step in full TOE certification. A Security Target claiming full compliance with the SCSUG-SCPP would be offered. The TOE would be the integrated platform as described in this PP. The evaluation evidence offered in support of the claims would be different from that offered in Figure D.1, however, since it should not be necessary to again submit all of the original information utilized in the evaluation of the integrated circuit ST and the operating software ST—that is, various components of the integrated TOE would have already been evaluated in a context which included precisely the same statements as those required in the SCSUG-SCPP. The SCSUG-SCPP evaluation could then refer to that supporting information for the basic component inputs. The only original evaluation that would be required would consist of those parts of the SCSUG-SCPP that depend on the integrated operation of the composite TOE.

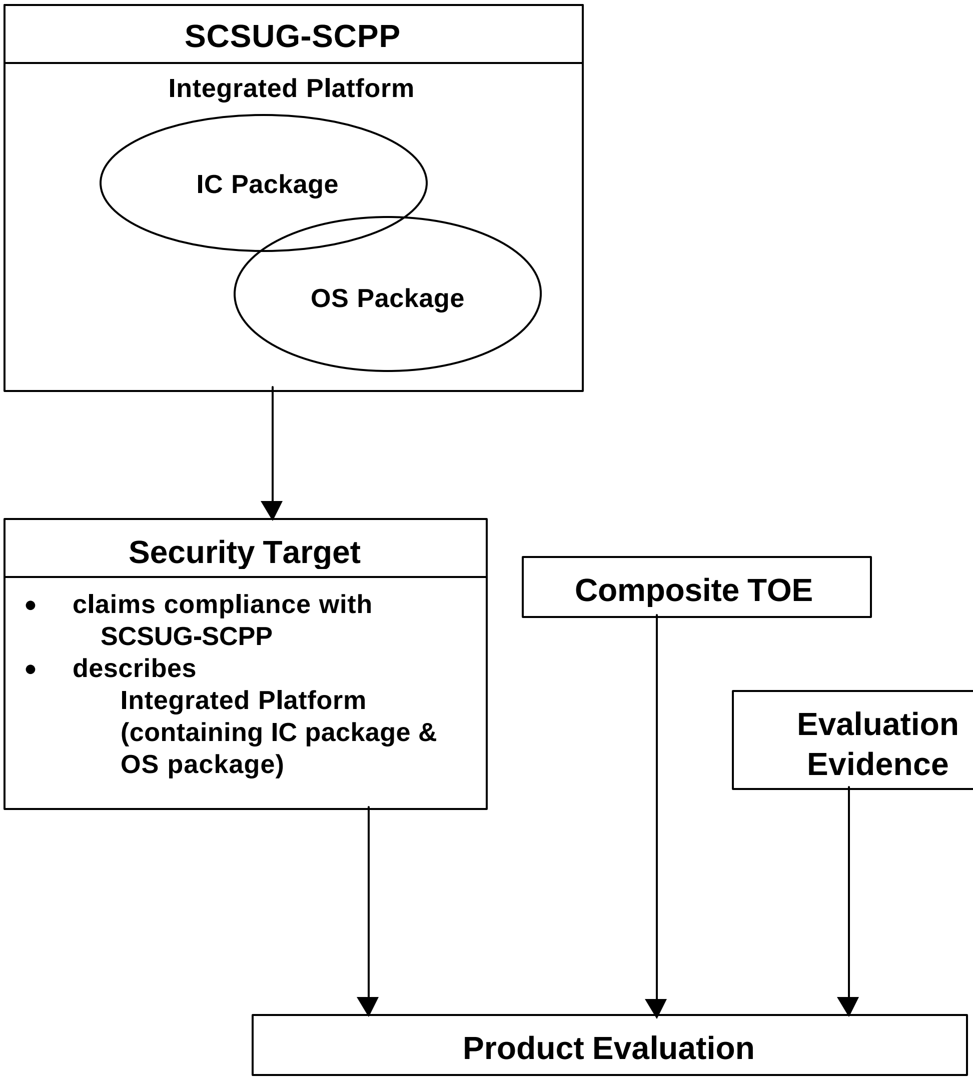

Figure D.1 SCSUG-SCPP Product Evaluation

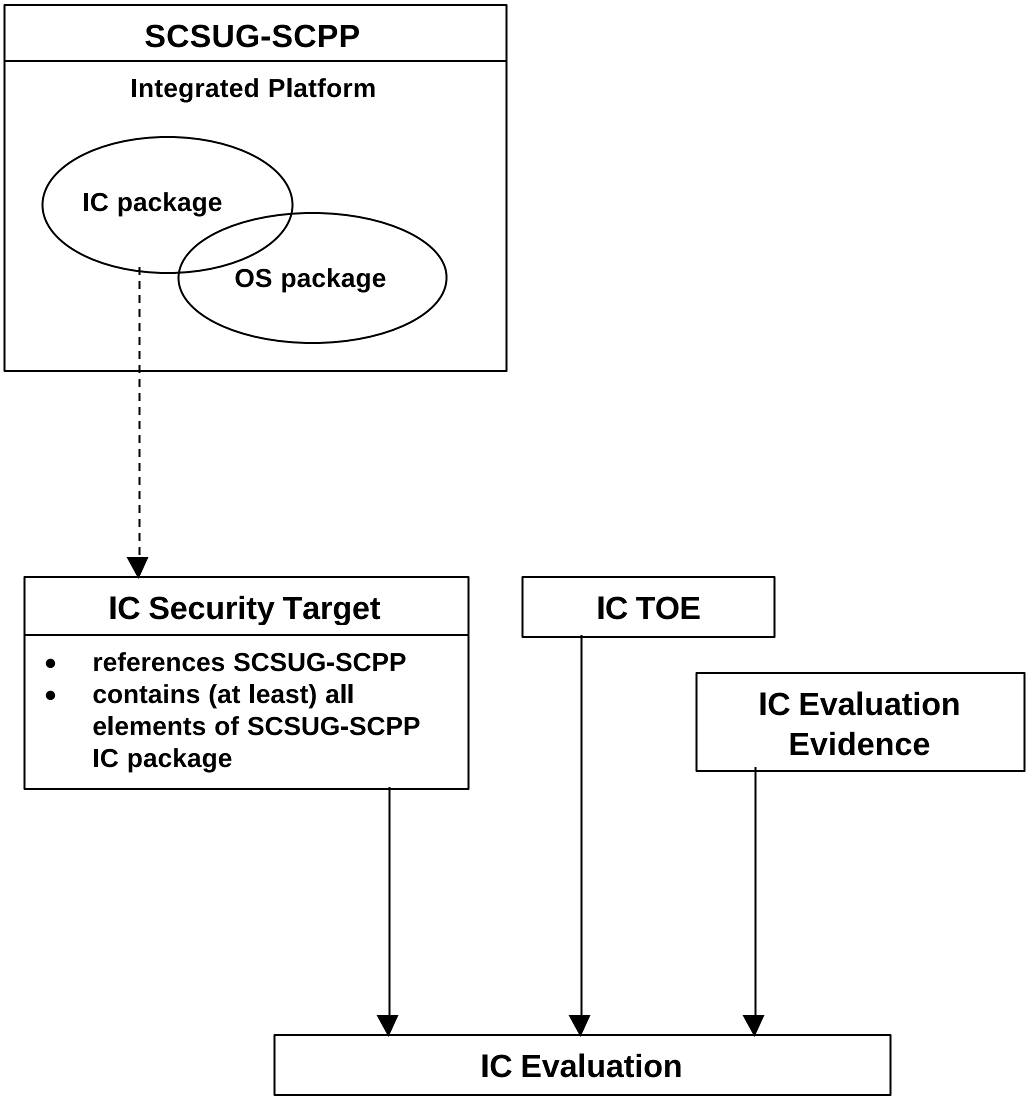

Figure D.2 IC Evaluation

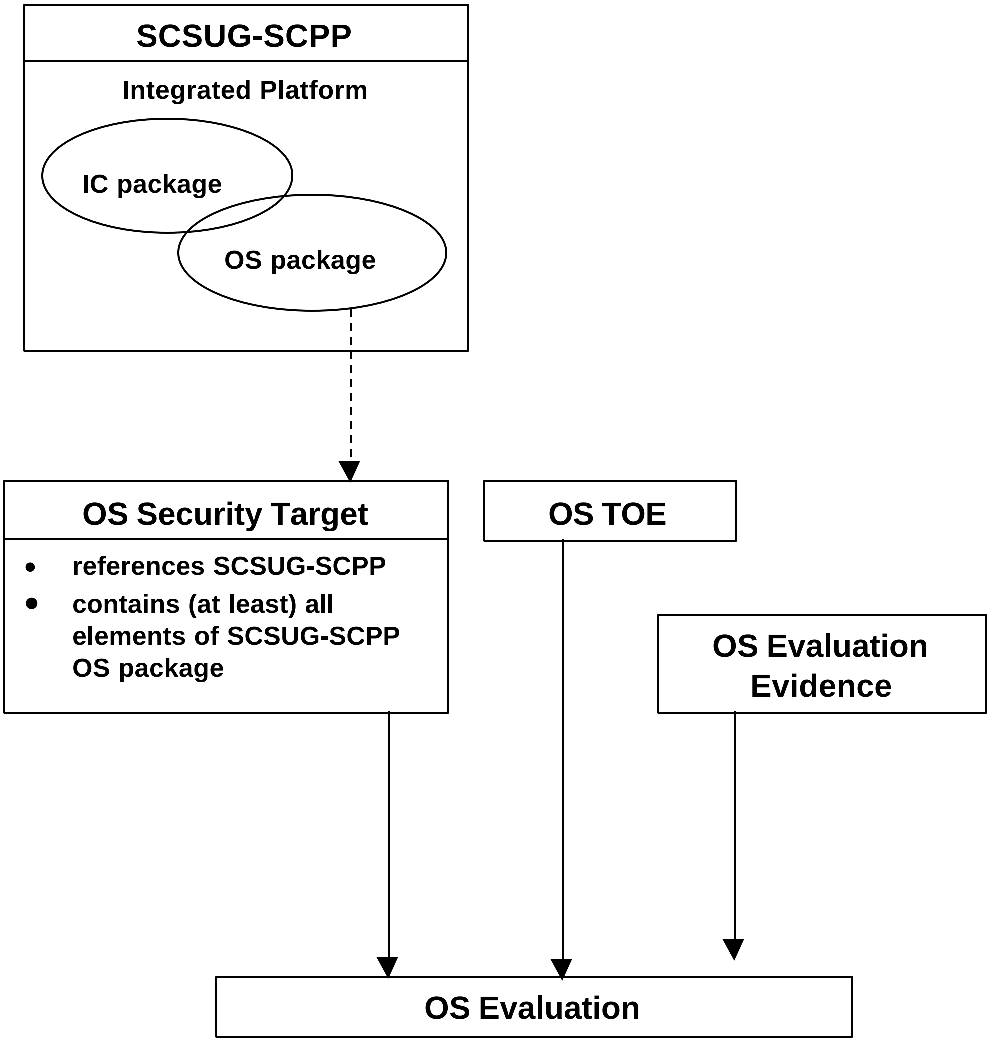

Figure D.3 OS Evaluation

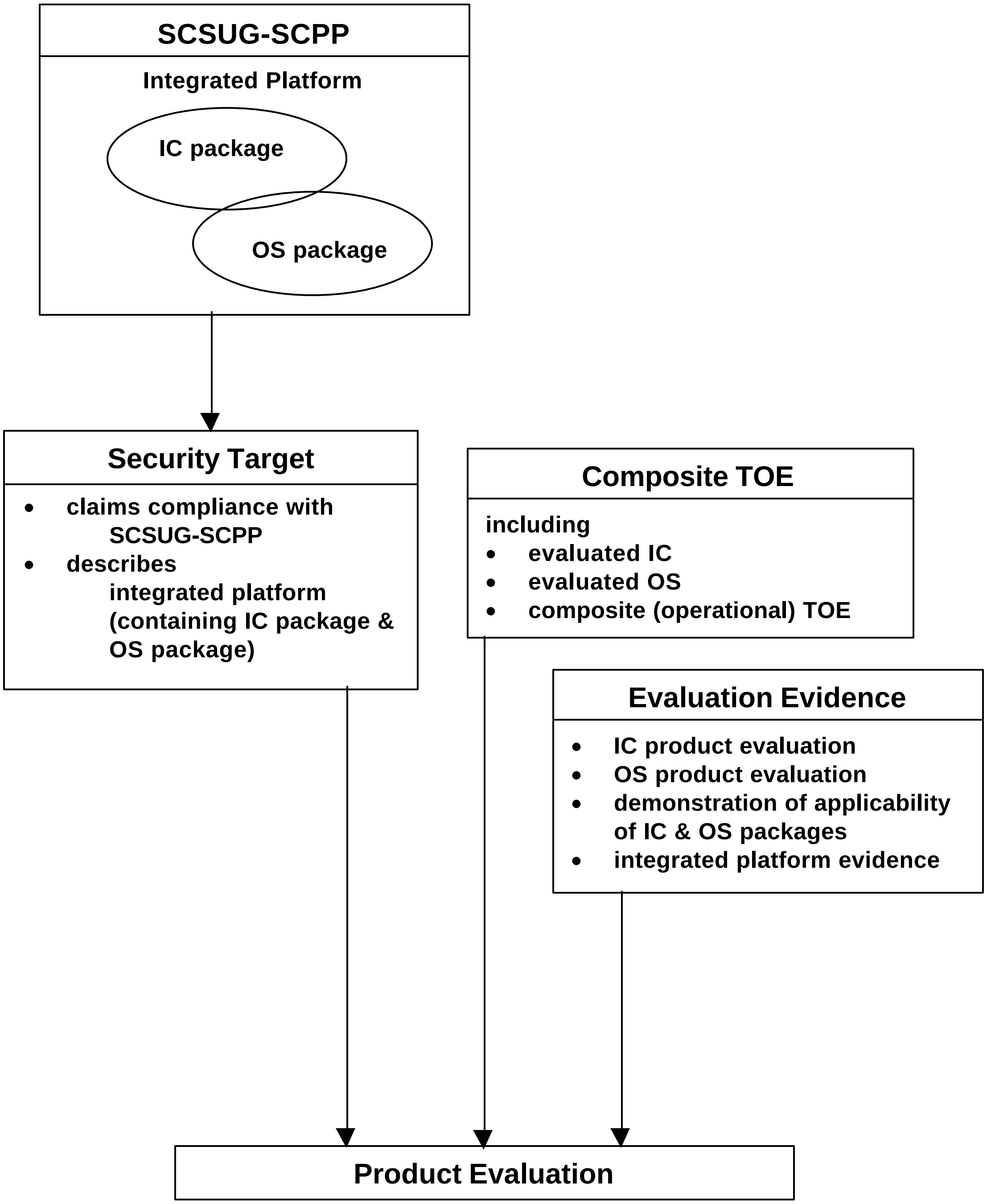
SCSUG-SCPP
Integrated Platform
IC package
OS package
Security Target
• claims compliance with SCSUG-SCPP
• describes integrated platform (containing IC package & OS package)
Composite TOE
including
• evaluated IC
• evaluated OS
• composite (operational) TOE
Evaluation Evidence
• IC product evaluation
• OS product evaluation
• demonstration of applicability of IC & OS packages
• integrated platform evidence
Product Evaluation

Figure D.4 SCSUG-SCPP Evaluation with Use of Packages

D.5 Package Construction

In the discussion of the packages below, note that some security environment statements (threats and policies), objectives, and requirements apply uniquely to one package, while some apply to multiple packages. The division into packages does, however, add a slightly differing emphasis to the various elements from that represented in the SCSUG-SCPP. This difference in emphasis derives from the detailed impact of the threats and policies being countered in each package. Accordingly, a short discussion is offered with each, providing the rationale for assignment to specific packages. Other than these package-specific discussions, the material detailing each package parallels the presentation in the main body of this protection profile, providing statements regarding the security environment, security objectives, and security requirements. Full definitions, descriptive text, information on rationales, and supporting discussions for each entry are contained in the main body of this PP.

D.5.1 Allocation of Threats to Packages

Table D.1 provides a listing of all of the threats described in the SCSUG-SCPP with an indication of how these should be associated with packages. This table is followed by a discussion of each threat element of the SCSUG-SCPP security environment, identifying why it is appropriate to identify that element with the specified packages. Again, it is to be noted that the integrated platform is synonymous with the SCSUG-SCPP itself. All elements apply and must be ultimately evaluated. Some elements are, however, particularly identified with synergistic effects between hardware and software and will require detailed evaluation in the final integrated product regardless of their use in an IC package or OS package. Those elements requiring special attention in the integrated platform evaluation are indicated by a special entry in the table and are discussed with each appropriate threat.

Table D.1 Package Threats

Threat	**Is Required In**		
	IC Package	**OS Package**	**Integrated Platform (see note)**
T.P_Probe	✓		✓
T.P_Modify	✓		✓
T.E_Manip	✓		☑

Threat	Is Required In		
	IC Package	**OS Package**	**Integrated Platform (see note)**
T.Flt_Ins		✓	✓
T.Forcd_Rst		✓	☑
T.Inv_Inp		✓	✓
T.Load_Mal		✓	✓
T.Reuse		✓	✓
T.Search		✓	✓
T.UA_Load		✓	✓
T.Access		✓	☑
T.First_Use		✓	✓
T.Impers		✓	✓
T.App_Ftn		✓	☑
T.LC_Ftn	✓	✓	☑
T.Res_Con		✓	✓
T.Crypt_Atk	✓	✓	☑
T.I_Leak	✓	✓	☑
T.Link		✓	☑
T.Env_Strs	✓	✓	☑
T.Lnk_Att	✓	✓	☑
T.Rep_Atk	✓	✓	☑
T.Clon	✓	✓	☑

Note: The Integrated Platform is the subject of the SCSUG-SCPP. As such, all threats must be addressed. Those marked ☑ represent synergistic effects between hardware and software and are of particular concern for evaluation in the integrated platform.

T.P_Probe (Physical Probing of the IC) deals with mechanical attacks on the structure of the integrated circuit. The integrated circuit construction determines the resistance of the TOE to this type of attack. It is therefore appropriate to include this threat primarily in the IC package.

T.P_Modify (Physical Modification) of the IC deals with attempts to physically modify the TOE such that information relating to the secure operation of the TOE is revealed. The integrated circuit construction determines the resistance of the TOE to this type of attack. It is therefore appropriate to include this threat primarily in the IC package.

T.E_Manip (Electrical Manipulation of the IC) addresses attempts in which the TOE is modified so that it can be directly fraudulently used. The integrated circuit design and implementation determine much of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the IC package. Some details which could contribute to vulnerability are functions of software. These are not purely design issues however, as they result from the interaction of hardware and software. They can not be checked through evaluation of the software alone, so this threat will require additional attention with the integrated platform.

T.Flt_Ins (Insertion of Faults) addresses the situation when the TOE is actively being probed through the deliberate insertion of selected inputs with the intent of observing the outputs. This is normally performed over multiple repetitions with small changes in the selected inputs. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Forcd_Rst (Forced Reset) addresses the situations in which the TOE is reset during operation. This may occur at any time including during a reset operation itself. The structure and implementation of the program being manipulated determine most of the response to this threat. It is therefore appropriate to include this threat in the OS package. The presence of hardware effects (such as logic races) dictates, however, that this function be further evaluated in the integrated platform.

T.Inv_Inp (Invalid Input) addresses the introduction of input which does not conform to the required style, content, or format. This input may have the look of accidental or erroneous entries (and that may be, in fact, the source of the data) but the result may be the misperformance of the TOE such that security is compromised. Attackers may use non-conforming data, existing but inappropriate commands, or well formatted commands with data requests that refer to locations which are outside of range or not to be utilized in that operation. The structure and implementation of the program receiving data determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Load_Mal (Data Loading Malfunction) addresses the situation in which an attack utilizes maliciously generated errors in the set-up information such that security can be compromised. This is related to T.UA_Load except that this threat deals with properly executed loading of corrupted information. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Reuse (Replay Attack) addresses attempts by an attacker to utilize the information available from a partially or fully completed operation to repeat the operation in a fraudulent fashion. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Search (Data Space Search) addresses the threat of an attacker's gaining knowledge of secure information through use of read commands to repetitively search the data space to extract all stored information. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.UA_Load (Unauthorized Program Loading) addresses the use of unauthorized programs that either exist in the TOE or are specifically loaded with the intent to penetrate the security features of the TOE. The structure and implementation of the program receiving these unauthorized loading functions determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Access (Invalid Access) addresses the need for unauthorized access to information or resources. This threat is distinguished by the emphasis on access of users to information. This is related to P.Data_Acc and is differentiated from P.File_Acc by its relation to data as opposed to file structures. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included in the OS package. In the event that the hardware may provide additional support through such structures as memory management units, this should be further evaluated in the integrated platform.

T.First_Use (Fraud on First Use) deals with fraud perpetrated through the use of smart cards which have not been officially issued. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Impers (Impersonation) addresses the use of the TOE by an attacker impersonating an authorized user. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.App_Ftn (Use of Unallowed Application Functions) deals with the exploitation of inappropriate interaction of functions between applications. The structure and implementation of the program being manipulated determine most of the response to this threat. Therefore, this threat is appropriately included in the OS package. The potential presence of different pieces of software in the final composite TOE dictates, however, that this function be further evaluated in the integrated platform.

T.LC_Ftn (Use of Unallowed Life Cycle Functions) deals with the exploitation of inappropriate interaction of functions between life cycle operations. The integrated circuit design and implementation determine some of the resistance of the TOE to this type of attack, so it is appropriate to include this threat in the IC package. The structure and implementation of the program being manipulated determine most of the response to this threat. Therefore, this threat is appropriately included in the OS package. The potential presence of different pieces of software and their interaction with the IC life cycle functions in the final composite TOE dictate, however, that this function be further evaluated in the integrated platform.

T.Res_Con (Resource Contention) addresses the utilization of an excessive amount of memory, program space, or other resource by a negligent user or an attacker, precluding further normal use of the TOE. The structure and implementation of the program being manipulated determine the response to this threat. Therefore, this threat is appropriately included primarily in the OS package.

T.Crypt_Atk (Cryptographic Attack) addresses direct attacks on the cryptographic mechanisms employed in the TOE. The integrated circuit design and implementation determine some of the resistance of the TOE to this type of attack, so it is appropriate to include this threat in the IC package. The operating software design and implementation also determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the OS package. Other vulnerabilities can not be determined, however, until the software is installed on the IC. It is therefore appropriate to specifically evaluate the integrated platform against this threat.

T.I_Leak (Information Leakage) deals with the exploitation of information inadvertently available from emanations or variations in power consumption or other operating parameters as a function of operation being performed. SPA and DPA are examples of such information leakage. The integrated circuit design and implementation determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the IC package. Operating software design and implementation also determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the OS package. Other vulnerabilities can not be determined, however, until the software is installed on the IC. It is therefore appropriate to specifically evaluate the integrated platform against this threat.

T.Link (Linkage of Multiple Observations) addresses the observation and linking of a variety of operations, leading to the attacker's ability to deduce useful information. This threat is differentiated from T.LC_Ftn and T.App_Ftn in that it entails purely observation of normally visible operations and not the manipulation entailed in using operations across defined boundaries. The structure and implementation of the program being manipulated determine most of the response to this threat. Therefore, this threat is appropriately included in the OS package. The potential presence of different pieces of software in the final composite TOE dictate, however, that this function be further evaluated in the integrated platform.

T.Env_Strs (Environmental Stress) deals with the imposition of environmental extremes on the TOE with the intent to cause a direct or indirect failure in the security mechanisms. The integrated circuit design and implementation determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the IC package. The operating software design and implementation also determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the OS package. Other vulnerabilities can not be determined, however, until the software is installed on the IC. It is therefore appropriate to specifically evaluate the integrated platform against this threat.

T.Lnk_Att (Linked Attacks) deals with multiple attacks synergistically causing a degradation and failure of TOE security. The integrated circuit design and implementation determine some of the response of the TOE to this type of attack so it is appropriate to include this threat in the IC package. The operating software design and implementation also determine some of the resistance of the TOE to this type of attack, so it is appropriate to include this threat in the OS package. Other vulnerabilities can not be determined, however, until the software is installed on the IC. It is therefore appropriate to specifically evaluate the integrated platform against this threat.

T.Rep_Atk (Audit Failure) represents the implicit threat of non-detection of attacks from other threats. Some of the potential inputs to this function are derived from the functions contained in the integrated circuit. Therefore, this threat is appropriately included in the IC package. The structure and implementation of the program being manipulated determine most of the response to this threat. Therefore, this threat is appropriately included in the OS package. The interaction of hardware and software effects regarding the input to the audit function dictates, however, that this function be further evaluated in the integrated platform.

T.Clon (Cloning) represents the threat that an attacker may manufacture all or a usable portion of the TOE. The integrated circuit design and implementation determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the IC package. The operating software design and implementation also determine some of the resistance of the TOE to this type of attack so it is appropriate to include this threat in the OS package. Other vulnerabilities can not be determined, however, until the software is installed on the IC. It is therefore appropriate to specifically evaluate the integrated platform against this threat.

D.5.2 Allocation of Policies to Packages

Table D.2 provides a listing of all of the policies described in the SCSUG-SCPP with an indication of how these should be associated with packages. This table is followed by a discussion of each policy of the SCSUG-SCPP security environment, identifying why it is appropriate to identify that element with the specified packages. Again, it is to be noted that the integrated platform is synonymous with the SCSUG-SCPP itself. All elements apply and must be ultimately evaluated through use of the SCSUG-SCPP. Some elements are, however, particularly identified with synergistic effects between hardware and software and will require evaluation in the final integrated product. These are indicated by a special entry in the table and are discussed with each appropriate policy.

Table D.2 Package Policies

Policy	**Is Required In**		
	IC Package	**OS Package**	**Integrated Platform (see note)**
P.Crypt_Std	✓	✓	☑
P.Data_Acc		✓	✓
P.File_Acc		✓	✓
P.Ident	✓	✓	☑
P.Sec_Com	✓	✓	☑
Note: The Integrated Platform is the subject of the SCSUG-SCPP. As such, all policies must be addressed. Those marked ☑ represent synergistic effects between hardware and software and are of particular concern for evaluation in the integrated platform.			

P.Crypt_Std (Cryptographic Standards) establishes that accepted cryptographic standards be used in the design of the TOE. The implementation of the IC and of the OS must conform to this usage. It is therefore appropriate to include this policy in both the IC package and the OS package. Further, the implementation of the integrated platform contributes to the instantiation of cryptologic mechanisms. This policy therefore requires attention with the integrated platform.

P.Data_Acc (Data Access) establishes that there must be a stated policy for access to data and data objects. This is differentiated from P.File_Acc by its relation to data as opposed to file structures. The implementation of the software determines the data policies for the TOE. Therefore, this policy is appropriately included primarily in the OS package.

P.File_Acc (File Access) establishes that there must be a stated policy for the right to establish files and file structures. This is differentiated from P.Data_Acc by its relation to file structures as opposed to data. The implementation of the software determines the file control policies for the TOE. Therefore, this policy is appropriately included primarily in the OS package.

P.Ident (Identification) establishes that there must be a clear, complete, and unique identification for the TOE. The implementation of both the IC and the OS must contribute to this unambiguous identification. It is therefore appropriate to include this policy in the IC package and in the OS package. Further, the implementation of the integrated platform must contribute to this unambiguous identification. This policy therefore requires attention with the integrated platform.

P.Sec_Com (Secure Communications) establishes that there is a secure communication channel between the TOE and the card acceptor device. The implementation of the IC and of the OS must contribute to this combination capability. It is therefore appropriate to include this policy in both the IC package and the OS package. The integration of functions in the TOE must also contribute to this communication capability. This policy therefore requires attention with the integrated platform.

D.5.3 Allocation of Objectives to Packages

The identification of the package environment statements leads directly to the determination of the security objectives. These are listed below in Table D.3. This table is followed by a discussion of each objective included in the SCSUG-SCPP, identifying why it is appropriate to list that element with the specified package. Full descriptions and detailed rationales are included in the main text of this protection profile. It should be noted that the specific emphasis imposed on the selected objective (e.g., as applying primarily to the IC, primarily to the OS, or mutually to the IC and OS) results in some slightly different assignment of objectives to environment statements than those in the SCSUG-SCPP itself. Thus, assembling the objectives for each of the individual packages may not provide a complete set of objectives. Additional entries will be required to generate a fully complete ST. The information for the Integrated Platform will be complete as it is described in the main text of this PP.

Table D.3 Package Objectives

Objective	**Is Required In**		
	IC Package	**OS Package**	**Integrated Platform (see note)**
O.Audit	✓	✓	☑
O.Crypt	✓	✓	☑
O.D_Read	✓	✓	✓
O.DAC		✓	✓
O.Env_Strs	✓	✓	☑
O.FAC		✓	✓
O.Flt_Ins		✓	✓
O.I_Leak	✓	✓	☑
O.Ident	✓	✓	☑
O.Init		✓	✓

Objective	**Is Required In**		
	IC Package	**OS Package**	**Integrated Platform (see note)**
O.Life_Cycle		✓	☑
O.Log_Prot		✓	☑
O.Mult_App		✓	☑
O.Phys_Prot	✓		✓
O.Res_Access		✓	✓
O.Reuse		✓	✓
O.Search		✓	✓
O.Sec_Com	✓	✓	☑
O.Set_Up		✓	✓
O.Unlink		✓	☑
Note: The Integrated Platform is the subject of the SCSUG-SCPP. As such, all objectives must be addressed. Those marked ☑ represent synergistic effects between hardware and software and are of particular concern for evaluation in the integrated platform.			

O.Audit (Audit) ensures that some specified data is recorded and available for analysis such that the nature of repetitive attacks may be determined and countered. The appropriate response for this objective is determined in the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so this objective also requires specific attention in the integrated platform.

O.Crypt (Cryptography) ensures that any cryptographic functions available are performed in a secure manner. This is determined in the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so this objective also requires specific attention in the integrated platform.

O.D_Read (Data Read Format) ensures that data available on data busses inside the TOE provides no information beyond that which would be available through statically reading the memory. This is determined in the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package.

O.DAC (Data Access Control) (in conjunction with definitions included in FDP_ACF.1) establishes the data access policies. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Env_Strs (Environmental Stress) ensures that the TOE performs in an acceptable fashion (i.e., does not reveal secure information) when exposed to out of design specification conditions. This is determined in the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so this objective also requires specific attention in the integrated platform.

O.FAC (File Access Control (in conjunction with definitions included in FDP_IFF.1) establishes the file control policies. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Flt_Ins (Fault Insertion) ensures that active probing of the TOE through the deliberate insertion of selected inputs with the intent of observing the outputs is resisted. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.I_Leak (Information Leakage) addresses the issue of the exploitation of information inadvertently available from emanations or variations in power consumption or other operating parameters as a function of the operation being performed. SPA and DPA are examples of such information leakage. The appropriate response for this objective is determined in the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so this objective also requires specific attention in the integrated platform.

O.Ident (TOE Identification) ensures that there is a clear, complete, and unique identification for the TOE. Such identification must be applied during development of both the IC and the OS and this objective is therefore appropriate for inclusion in both the IC package and the OS package. The final product requires identification so it is also appropriate to specifically evaluate the integrated platform against this objective.

O.Init (Initialization) ensures that the TOE always enters its defined initial state upon reset. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Life_Cycle (Life Cycle Functions) ensures that the exploitation of inappropriate interaction of functions between different life-cycle operations does not compromise security through unauthorized availability of information. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion in the OS package. The potential presence of different pieces of software in the final composite TOE dictates, however, that this function be further evaluated in the integrated platform.

O.Log_Prot (Logical Protection) ensures that the TOE is constructed such that it responds in a secure manner to all probing represented by data, commands, or other input which is not fully conforming to the anticipated style and content. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion in the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so this objective also requires specific attention in the integrated platform.

O.Mult_App (Multiple Applications) ensures that the exploitation of inappropriate interaction of functions between operations and applications does not compromise security through unauthorized availability of information. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion in the OS package. The potential presence of different pieces of software in the final composite TOE dictates, however, that this function be further evaluated in the integrated platform.

O.Phys_Prot (Physical Protection) ensures that the TOE is constructed using such elements as protective layering, special rules regarding integrated circuit layout, and removal of test pads after initial (wafer) testing is complete. These actions are intended to make it difficult to derive information from the IC and, if such information is derived, to make it difficult to interpret and apply such information to attempts to compromise. This objective is appropriate for the IC package.

O.Res_Access (Resource Access) prevents the monopolization of resources by a single entity. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Reuse (Replay) ensures that no assets can be compromised in the event of an attempt by an attacker to utilize the information available from a partially or fully completed operation to repeat the operation in a fraudulent fashion replay. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Search (Data Search) prevents repeated entry to data spaces which may be subject to search. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Sec_Com (Secure Communications) ensures that the TOE is capable of establishing and using a secure communication channel between the TOE and the card acceptor device. This is determined in the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some of the required capability depends on the interaction between hardware and software however, so this objective also requires specific attention in the integrated platform.

O.Set_Up (Set-Up Sequence) ensures that a defined and controlled sequence of events is completed before the TOE is enabled for use. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

O.Unlink (Linkage) ensures that information exposed in each individual operation is not of use to an attacker in understanding and attacking the TOE. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion in the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so this objective also requires specific attention in the integrated platform.

D.5.4 Allocation of Requirements to Packages

The identification of the package objectives leads directly to the determination of the security requirements. The security functional requirements are listed below in Table D.4 This table is followed by a discussion of each security functional requirement included in the SCSUG-SCPP, identifying why it is appropriate to list that element with the specified package. Table D.5 follows with the related discussions for each security assurance requirement. Table D.5 only references those security assurance requirements explicitly required to meet specific objectives. Additional requirements, consistent with an evaluation assurance level 4 will also be necessary as discussed in Section D.6. Full descriptions and detailed rationales are included in the main text of this protection profile.

It should be noted that the specific emphasis imposed on the selected requirements (e.g., as applying primarily to the IC, primarily to the OS, or mutually to the IC and OS) results in some slightly different assignment of requirements to objectives than that in the full SCSUG-SCPP. Thus, assembling the requirements for each of the individual packages may not provide a complete set of requirements. Additional entries may be required to generate a fully complete ST. The information for the integrated platform will be complete as it is described in the main text of this PP.

Table D.4 Package Security Functional Requirements

Requirement	Is Required In		
	IC Package	OS Package	Integrated Platform (see note)
FAU_ARP.1	✓	✓	✓
FAU_LST.1	✓	✓	✓
FAU_SAA.1		✓	✓
FAU_SEL.1	✓	✓	✓
FAU_STG.1	✓	✓	☑
FAU_STG.4		✓	✓
FCS_CKM.1	✓	✓	☑

Requirement	Is Required In		
	IC Package	OS Package	Integrated Platform (see note)
FCS_CKM.3	✓	✓	☑
FCS_COP.1	✓	✓	☑
FDP_ACC.1		✓	✓
FDP_ACF.1		✓	✓
FDP_ETC.1	✓	✓	☑
FDP_IFC.1		✓	✓
FDP_IFF.1		✓	✓
FDP_ITC.1	✓	✓	☑
FDP_ITT.1	✓	✓	☑
FDP_RIP.1		✓	✓
FDP_UIT.1	✓	✓	☑
FIA_AFL.1		✓	✓
FIA_ATD.1		✓	✓
FIA_UAU.1		✓	✓
FIA_UAU.7	✓	✓	☑
FIA_UID.1		✓	✓
FMT_MOF.1		✓	✓
FMT_MSA.1		✓	✓
FMT_MSA.2		✓	✓
FMT_MSA.3		✓	✓
FMT_MTD.1		✓	✓
FMT_MTD.2		✓	✓
FMT_MTD.3		✓	✓
FMT_REV.1		✓	✓
FPT_FLS.1	✓	✓	☑

Requirement	Is Required In		
	IC Package	OS Package	Integrated Platform (see note)
FPT_ITI.1	✓	✓	☑
FPT_ITT.1	✓	✓	☑
FPT_PHP.3	✓	✓	☑
FPT_RCV.3	✓	✓	☑
FPT_RCV.4	✓	✓	☑
FPT_RPL.1		✓	✓
FPT_RVM.1	✓	✓	☑
FPT_SEP.1	✓	✓	☑
FPT_TST.1	✓	✓	☑
FRU_RSA.1		✓	✓
FTP_ITC.1	✓	✓	☑
Note: The integrated platform is the subject of the SCSUG-SCPP. As such, all requirements must be addressed. Those marked ☑ represent synergistic effects between hardware and software and are of particular concern for evaluation in the integrated platform.			

FAU_ARP.1 (Security alarms) provides for a response when selected violations are noted. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package.

FAU_LST.1 (Audit list generation) provides for the generation of the information to be audited. This is primarily determined in the design and implementation of the OS and is therefore appropriate for inclusion in the OS package. The possible inputs for audit however, are derived from the design and implementation of the IC and therefore this requirement is appropriate for inclusion in the IC package.

FAU_SAA.1 (Potential violation analysis) provides selection of rules to monitor for potential violations. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FAU_SEL.1 (Selective audit) provides selection of audit information. This is primarily determined in the design and implementation of the OS and is therefore appropriate for inclusion in the OS package. The possible inputs for audit however, are derived from the design and implementation of the IC and therefore this requirement is appropriate for inclusion in the IC package.

FAU_STG.1 (Protected audit trail storage) provides protection of the audit data itself. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FAU_STG.4 (Prevention of audit data loss) provides the operations to be followed in case of overflow of audit information. This is primarily determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FCS_CKM.1 (Cryptographic key generation) provides for generation of keys in accordance with an accepted standard. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FCS_CKM.3 (Cryptographic key access) provides for secure key access in accordance with an accepted standard. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FCS_COP.1 (Cryptographic operation) provides for operation of cryptographic functions in accordance with an accepted standard. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FDP_ACC.1 (Subset access control) defines the access control policies and defines the scope of these policies. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FDP_ACF.1 (Security attribute based access control) provides the rules and enforcement for access to specified controlled subjects and objects. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FDP_ETC.1 (Export of user data without security attributes) provides the means of controlling the information which can be exchanged through imposition of the access control SFP and the information flow control SFP. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FDP_IFC.1 (Subset information flow control) defines the information flow control policies and defines the scope of these policies. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FDP_IFF.1 (Simple security attributes) provides the rules and enforcement for information flow between a controlled subject and controlled information via a controlled operation. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FDP_ITC.1 (Import of user data without security attributes) provides the means of controlling the information which can be exchanged through imposition of the access control SFP and the information flow control SFP. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FDP_ITT.1 (Basic internal transfer protection) provides the means of preventing the disclosure or modification of user data when it is transmitted between parts of the TOE according to the policies expressed in the access control SFP and the information flow control SFP. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FDP_RIP.1 (Subset residual information protection) provides for the protection of information when the resource containing that information is no longer in use. This provides protection to all but the immediately operating elements. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FDP_UIT.1 (Data exchange integrity) provides for user data exchange without modification. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FIA_AFL.1 (Authentication failure handling) ensures a limit on the number of authentication attempts which can be made. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FIA_ATD.1 (User attribute definition) provides the list of user security attributes. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FIA_UAU.1 (Timing of authentication) covers the requirements necessitating authentication. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FIA_UAU.7 (Protected authentication feedback) provides for the elimination of all feedback during authentication, removing that potential source of information from an attacker. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FIA_UID.1 (Timing of identification) provides the requirements for which actions can be taken prior to imposition of identification. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MOF.1 (Management of security functions behavior) allows the authorized roles to manage the behavior of functions in the TSF. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MSA.1 (Management of security attributes) allows authorized roles to manage the security attributes. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MSA.2 (Secure security attributes) establishes that only secure values can be input for security attributes. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MSA.3 (Static attribute initialization) provides the restrictive initial attributes and default values. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MTD.1 (Management of TSF data) allows authorized roles to manage specified TSF data. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MTD.2 (Management of limits on TSF data) allows the management of which limits can be set and which roles are allowed to perform that action. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_MTD.3 (Secure TSF data) establishes that only secure values are accepted for TSF data. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FMT_REV.1 (Revocation) identifies the roles that are allowed to revoke the security attributes necessary to have access. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FPT_FLS.1 (Failure with preservation of secure state) provides for acceptably secure operation in the event of failures. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_ITI.1 (Inter-TSF detection of modification) provides for TSF data exchange without modification. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_ITT.1 (Basic internal TSF data transfer protection) specifically protects TSF data from modification. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_PHP.3 (Resistance to physical attack) provides for features that resist physical tampering with the TSF elements. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_RCV.3 (Automated recovery without undue loss) provides for acceptably secure operation in the event of failures through automated recovery, preventing undue loss of protected objects. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_RCV.4 (Function recovery) provides for acceptably secure operation in the event of failures through requiring either successful completion of operations or rollback to a previously defined state. The instance of power failure is of particular concern due to the stated unreliability of supply. Satisfaction of this requirement is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_RPL.1 (Replay detection) ensures that replay is detected for specified entities and that a specified action is taken in response. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FPT_RVM.1 (Non-bypassability of the TSP) provides the necessary separation and protection to the TSF so that the required TSPs can be successfully applied. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_SEP.1 (TSF domain separation) provides the necessary separation and protection to the TSF so that the required TSPs can be successfully applied. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FPT_TST.1 (TSF testing) generates the initial self-test verifying that the TSF is operating correctly. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

FRU_RSA.1 (Maximum Quotas) establishes the maximum limits for a resource which may be dedicated to users and/or subjects to prevent inappropriate allocations to deny further service. This is determined in the design and implementation of the OS and is therefore appropriate for inclusion primarily in the OS package.

FTP_ITC.1 (Inter-TSF trusted channel) provides the establishment of a trusted channel. This is derived from the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some vulnerabilities may be introduced through the synergistic actions between hardware and software however, so it is also necessary to provide specific attention to this requirement in the integrated platform.

Table D.5 Package Security Assurance Requirements

Requirement	Is Required In		
	IC Package	OS Package	Integrated Platform (see note)
ACM_CAP.4	✓	✓	☑
ADV_IMP.1	✓	✓	☑
ADV_INT.1	✓	✓	✓
AVA_VLA.3	✓	✓	☑
Note: The integrated platform is the subject of the SCSUG-SCPP. As such, all requirements must be addressed. Those marked ☑ represent synergistic effects between hardware and software and are of particular concern for evaluation in the integrated platform.			

ACM_CAP.4 (Generation support and acceptance procedures) requires the developer to describe and maintain the methods used to uniquely identify the configuration items, including all parts of the TOE. This is important during the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some additional configuration elements may be introduced during integration, so it is also necessary to provide specific attention to this requirement in the integrated platform.

ADV_IMP.1 (Subset of the implementation of the TSF) provides for the review and evaluation of selected subsets of the TOE implementation. This is important during the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some additional vulnerabilities may be introduced during integration, so it is also necessary to provide specific attention to this requirement in the integrated platform.

ADV_INT.1 (Modularity) ensures that the developer shall use modular design for the structure of the TSF. This is important during the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package.

AVA_VLA.3 (Moderately resistant) provides for the review of identified vulnerabilities. This is important during the design and implementation of both the IC and the OS and is therefore appropriate for inclusion in both the IC package and the OS package. Some additional vulnerabilities may be introduced during integration, so it is also necessary to provide specific attention to this requirement in the integrated platform.

D.6 Package Completion Requirements

The assignments discussed above are based strictly on an allocation of the related SCSUG-SCPP TOE environment statements as they might apply to the generic separation into packages, describing the integrated circuit, operating software, and integrated platform. The information presented with the assignments deals with the justification for the assignment into that package. As suggested on page 71 of ISO 15546, Information Technology – Security Techniques – Guide for the Production of Protection Profiles and Security Targets:

> In order to be useful, a functional package must be reusable in a larger functional package, or in a PP or ST. A PP or ST author is likely to find the following information helpful:
>
> a) an identification of the security objectives which the SFRs satisfy
>
> b) notes on the use of ISO/IEC 15408 Part 2 components, or on the deviation from ISO/IEC 15408 Part 2
>
> c) rationale for the SFRs, covering:
> - the suitability of the SFRs to satisfy the identified security objectives
> - dependency analysis
> - demonstration of mutual support between SFRs

Thus, the packages, as presented above, represent only a partial definition for a full functional package. Most of the information necessary to complete specification of the packages is contained in the main text of this PP. It would, of necessity, be tailored to the specific nature of the packages being constructed.

The package definitions, as discussed above, do not address assurance level. This would be assumed to be EAL4, augmented, as appropriate. This is required so that any evaluation performed on a TOE which uses these packages as part of a security target would be completed at a level no lower than that required for the evaluation of an SCSUG-SCPP compliant TOE.

It should also be noted that these package definitions do not constitute sufficient information for direct generation of a security target. They do not address requirements on the TOE environment, assumptions, additional dependencies, etc. Additional information will need to be added to complete the specification of the appropriate component as required in an ST.

Annex E - Points of Contact

Smart Card Security User Group

American Express

10030 North 25th Avenue
MC26-03-03/TRC-D
Phoenix, AZ 85021 USA
Mark Merkow; e-mail <mark.merkow@aexp.com>

Europay International

Chaussėe de Tervuren 198A
B-1410 Waterloo
Belgium
Marijke de Soete; e-mail <mds@europay.com>

JCB Co Ltd

1-6 Kanda Surugadai, Chiyoda-Ku
Tokyo, 101-8006, Japan
Masanori Maeda; e-mail <maeda@cp.jcb.co.jp>

MasterCard International

2000 Purchase Street
Purchase, NY 10577-2509 USA
Terry Stanley; e-mail <terry_Stanley@mastercard.com>

Mondex International

47-53 Cannon Street
London EC4M 5SQ
England
Ken Warren; e-mail <ken.warren@mondex.com>

Visa International

Post Office Box 8999
San Francisco, CA 94128-8999 USA
Ken Ayer; e-mail <kayer@visa.com>

NIST

United States Department of Commerce
National Institute of Standards and Technology
Gaithersburg, MD 20899-0001 USA
Gene Troy; e-mail <eugene.troy@nist.gov>

NSA

National Security Agency
9800 Savage Road, Suite 6713
Fort George G. Meade, MD 20755-6713 USA
Stu Katzke; e-mail <swkatzk@missi.ncsc.mil>

RMTCI

Ray-McGovern Technical Consultants, Inc.
22304 East 67th Street
Broken Arrow, OK 74014-6621 USA
Douglas E. McGovern; e-mail <demcgovern@aol.com>

CC Management Committee Representatives to SCSUG

Australia

DSD-Australia
Locked Bag 5076
Kingston, ACT, 2604, Australia
Peter Lilley; e-mail <peter.lilley@dsd.gov.au>

Canada

Communications Security Establishment
PO Box 9703 Terminal
Ottawa, Ontario K1G 3Z4, Canada
Gerald Rose; e-mail <gdrose@its.cse.dnd.ca>

France

Information Technology Security Certification Centre, SCSSI
18, rue du Docteur Zamenhof
F092131 Issy-les-Molineaux, Cedex, France
Carlos Martin; e-mail <martincarlos@compuserve.com>

Germany

BSI/GISA
Godesberger Allee 183
Postfach 20-03-63
D-53133 Bonn, Germany
Dr. Hartwig Kreutz; e-mail <hkr@bsi.de>

United Kingdom

Communications-Electronics Security Group
P.0. Box 152
Cheltenham, Glos, GL52 5UF, England
Alan Borrett; e-mail<alan_borrett@cesg.gov.uk>

United States of America

National Institute of Standards and Technology
100 Bureau Drive, MS: 8930
Gaithersburg, MD 20899-8930, USA
Dr. Ron Ross; e-mail <rross@nist.gov>

National Security Agency
9800 Savage Road, Suite 6713
Fort Meade, MD 20755-6713, USA
Dr. Stu Katzke; e-mail <swkatzk@missi.ncsc.mil>